VŠB-Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2014 Bc. Miroslav Valečko

VŠB-Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Analýza dat pomocí Business Intelligence v SQL Serveru
Data Analysis by SQL Server Business Intelligence

2014 Bc. Miroslav Valečko

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Miroslav Valečko**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Analýza dat pomocí Business Intelligence v SQL Serveru
Data Analysis by SQL Server Business Intelligence

Zásady pro vypracování:

Microsoft SQL Server obsahuje ve vyšších edicích prostřednictvím rozšíření Business Intelligence (BI) podporu pro dolování dat.
Diplomant v rámci diplomové práce prostuduje metody, které jsou v tomto nástroji obsaženy a na vhodných datových kolekcích provede experimenty srovnávající vybrané metody pro analýzu dat obsažené v BI.

Jednotlivé body práce jsou:
1. Nastudování možností v BI.
2. Popis vybraných metod pro analýzu dat obsažených v BI.
3. Popsání postupu jak pomocí BI zpracovat experiment a to od surových dat, až pro analýzu a extrakci výstupů pomocí SQL dotazů.
4. Prostudování speciálního algoritmu obsaženého v BI pro zpracování sekvenčních dat.
5. Provedení experimentů vybranými metodami pro analýzu dat obsaženými v BI a jejich zhodnocení.

Seznam doporučené odborné literatury:

Ľuboslav Lacko: Business Intelligence v SQL Serveru 2008, COMPUTER PRESS, 2009

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jan Martinovič, Ph.D.**

Datum zadání: 01.09.2013
Datum odevzdání: 07.05.2014

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum: 2. 5. 2014 Podpis: Valečko

Poděkování

Velice rád bych poděkoval svému vedoucímu, Ing. Janu Martinovičovi, Ph.D., za cenné rady a odbornou pomoc, díky které jsem byl schopen vytvořit tuto diplomovou práci.

Abstrakt

Tato diplomová práce se zabývá dolováním dat pomocí Microsoft SQL Server 2012. Business Inteligence je rozsáhlé odvětví, do kterého spadá také problematika dolování dat. Dolování dat nám umožnuje získávat velice cenné a důležité informace. Toto odvětví je velice důležité v informačních technologiích, pomocí kterých můžeme provádět různé experimenty nad daty. V této práci se nejprve zaměříme na vybrané algoritmy pro dolování dat, které jsou obsaženy v Microsoft SQL Server 2012 Analysis Services. Dále si vysvětlíme, jak vybrané algoritmy pracují, včetně parametrů které obsahují. Také si řekneme, jak pomocí těchto parametrů můžeme ovlivňovat výsledné modely pro dolování dat. Pak si ukážeme jak provádět analýzu výsledných modelů pro dolování dat. Porovnáme jednotlivé výsledky, kdy v prvním případě necháme veškeré parametry ve standartním nastavení a poté upravíme parametry jednotlivých algoritmů, abychom dosáhli lepších výsledků.

Klíčová slova

dolování dat, trénování modelů, data, parametry, výsledky, předpovědi, doporučení

Abstract

Thesis is focused on a data acquisition by using Microsoft SQL Server 2012. Business intelligence is an extensive area and data acquisition is a part of this issue. Data acquisition allows obtaining of valuable and important information. In area of information technologies where is possible to realize various experiments with data is data acquisition very important. First part of this thesis is focused on a selected algorithms used for data acquisition which are contained in Microsoft SQL Server 2012 Analysis Services. Subsequently is explained how selected algorithms works including parameters that contains. It is also explained how is possible to influence resultant model for data acquisition by mentioned parameters. Then is shown how to analyze models for a data acquisition. In thesis is comparison of individual results. In first case all parameters are in a standard adjustment. Then parameters of individual algorithms are adjusted to achieve the best results.

Keywords

data mining, data, training models, parameters, results, predictions, recommendations

Seznam použitých zkratek a symbolů

KDD – Knowledge Discovery in Databases, dobývání znalostí z databáze
MAA – Microsoft Association Algorithm
MCA – Microsoft Clustering Algorithm
MDTA – Microsoft Decision Trees Algorithm,
MLRA – Microsoft Linear Regresion Algorithm
MNBA – Microsoft Naive Bayes Algorithm,
MNNA – Microsoft Neural Network Algorithm
MSCA – Microsoft Sequence Clustering Algorithm
MTSA – Microsoft Time Series Algorithm

Obsah

1. Úvod

V dnešní době jsou velice rozšířeny podnikové databázové systémy, ve kterých si jednotlivé firmy či korporace uchovávají pro ně velice důležité informace. Databázové informační systémy se nacházejí všude kolem nás, aniž bychom si to uvědomovali. Můžeme je najít například v lékařství, bankovnictví, v telekomunikacích, také se s nimi setkáváme při procházení jakýchkoliv webových stránek, a to především u internetových obchodů. Databázové tabulky obsahují miliony až miliardy záznamů, které jsou různě uspořádané a členěné [1].

Data Mining [1] neboli dolování dat je také označováno jako Knowledge Discovery in Databases (KDD). Dolování dat je součástí Business Inteligence, slouží pro analyzování dat, ve kterých se pokouší nacházet jednotlivé vzory v datech, předpovídat trendy, vytvářet doporučení a pravidla, ale také analyzovat sled událostí v datových sadách a získávat nové poznatky. Tyto vzory nemohou být objeveny normálním způsobem, protože se jedná o příliš složité vztahy, nebo máme příliš velký počet zkoumaných dat. Dolování dat patří k nejrychleji rostoucímu segmentu v Business inteligence. Pro rozpoznávání vzorů a trendů využívá matematické a statistické metody, ale také metody umělé inteligence. To umožnuje zlepšení obchodních a marketingových aktivit, možnost sledování a předvídání trendů, a tedy v konečném důsledku hlavně zvýšení konkurenceschopnosti firmy [1].

V této diplomové práci se zaměříme na nastudování vybraných algoritmů pro dolování dat, které jsou obsaženy v Microsoft SQL Server 2012 Analysis Services. Vysvětlíme si vybrané algoritmy a jejich specifikace na vstupní data. U vybraných algoritmů si ukážeme, jaké parametry obsahují, a popíšeme si význam jednotlivých parametrů. Pomocí těchto parametrů můžeme ovlivňovat výsledné modely pro dolování dat. Pochopení těchto parametrů je důležité pro vytváření přesnějších výsledných modelů. Dále jsou důležité pro ovlivňování výsledných modelů pro dolování dat. V neposlední řadě jsou tyto parametry velice důležité, jestliže vstupní tabulky obsahují velmi mnoho dat. Pomocí parametrů můžeme ovlivnit, kolik vstupních dat se použije pro vytváření výsledného modelu pro dolování dat. Tímto zamezíme přeučení výsledného modelu. Přeučený model poznáme tak, že na trénovacích datech je přesnost modelu vysoká, ale na nových datech je již nízká. Dále snížíme časovou náročnost a v neposlední řadě snížíme nároky na výpočetní prostředky daného počítače, na kterém se budou výsledné modely vypočítávat. Při studiích algoritmů bylo především čerpáno z Microsoft Library [2].

V práci se dále seznámíme s prostředím Microsoft Visual Studio 2012 s rozšířením pro Business Inteligence, díky kterému můžeme vytvářet modely pro dolování dat a následné vytváření analýz z jednotlivých modelů.

Projdeme si surová data, která jsou uložená v několika datových souborech. Pomocí naprogramované aplikace provedeme import dat do vytvořených tabulek.

Dále otestujeme vybrané algoritmy pro dolování dat na testovacích datech, poté provedeme analýzu jednotlivých výsledků a určíme úspěšnosti odhadů jednotlivých algoritmů nad testovanými daty. Pro každý algoritmus vytvoříme dva modely. Jeden model bude základní, kdy necháme všechny parametry v základním nastavení. Ve druhém modelu poupravíme parametry a výsledný model porovnáme se základním modelem. Určíme statistické úspěšnosti mezi těmito dvěma vytvořenými modely, kdy úpravou parametrů bychom měli dosáhnout lepších

výsledků. Jako poslední provedeme analýzu modelu pro sekvenční data, ve kterém provedeme analýzu vytvořených klastrů a jeho obsahu Názorně si ukážeme, proč jsou téměř stejné sekvence přiřazeny do jiného klastru.

1.1 Struktura práce

Představíme si co je to dolování dat v kapitole 2. Z jakých procesů se skládá dolování dat, uvedeme v kapitole 2.1. Mezi ně patří definice problému (kapitola 2.1.1), příprava dat (kapitola 2.1.2), zkoumání dat (kapitola 2.1.3), vytváření modelů pro dolování dat (kapitola 2.1.4), prozkoumávání a ověření modelu (kapitola 2.1.5), nasazení a aktualizace modelu (kapitola 2.1.6). V kapitolách 2.2 a 2.3 si řekneme, k čemu lze a nelze použít dolování dat. Ukážeme si, jaké algoritmy jsou obsaženy ve vyšších edicích Microsoft SQL Server 2012 v kapitole 2.4. Budeme se zabývat správným výběrem algoritmu v kapitole 2.5, podle typu dat (kapitola 2.5.1) a dle typu úlohy (kapitola 2.5.2).

V kapitole 3 se budeme zabývat popisem vybraných algoritmů. U každého vybraného algoritmu se v jeho podkapitolách dále věnujeme stručnému popisu algoritmu, jeho požadavkům na vstupní data a pomocí jakých parametrů lze nastavovat algoritmus. Jako první algoritmus je Microsoft Association Rules Algorithm v kapitole 3.1. Druhým algoritmem je Microsoft Clustering Algorithm v kapitole 3.2. Třetím algoritmem je Microsoft Decision Trees Algorithm v kapitole 3.3. Čtvrtým algoritmem je Microsoft Neural Network Algorithm v kapitole 3.4. Pátým a posledním algoritmem je Microsoft Sequence Clustering Algorithm v kapitole 3.5.

Testovacím datům, používaným v této práci se budeme věnovat v kapitole 4. Popíšeme si testovací data v kapitole 4.1 a poté se budeme věnovat jejich importu do databázových tabulek v kapitole 4.2.

V kapitole 5 si ukážeme jak pracovat s (programem) Microsoft Visual Studio 2012 s nainstalovaným rozšířením pro Business Intelligence. V několika krocích si řekneme jak založit a provést počáteční konfiguraci vytvořeného projektu, které je věnována kapitola 5.1. Ukážeme si nastavení a definování zdrojových dat (kapitola 5.1.1). Poté se budeme zabývat vytvořením modelu pro dolování dat v kapitole 5.2. Jak lze upravit model pro dolování dat si vysvětlíme v kapitole 5.3. Úpravy lze provádět pomocí datové struktury (kapitola 5.3.1) a parametrů (5.3.2).

Analýzu vytvořených modelů pro dolování dat provedeme v kapitole 6. V kapitole 6.1 se budeme postupně zabývat prohlížením sestavených modelů pro dolování dat. Odhadnutí přesnosti modelů pro dolování dat najdeme v kapitole 6.2. Odhadnutí lze provést také pomocí grafu, který nám vygeneruje Microsoft Visual Studio 2012. Poté si ukážeme, jak lze provádět vytváření nových odhadů v kapitole 6.3. V další kapitole 6.4 se budeme zabývat analýzou úspěšností odhadů modelů pro dolování dat. V poslední kapitole 6.5 provedeme analýzu modelu pro sekvenční data.

2. Dolování dat

Dnes si téměř většina firem uchovává veškeré informace v databázích. Tyto databázové systémy jsou různě členěné a uspořádané. V některých databázových systémech mohou být pouze stovky záznamů, v jiných dokonce až miliardy záznamů. V takovémto množství dat se mohou skrývat velice cenné a důležité informace. Informace mohou být různorodé, například číselné, znakové, textové, dvouhodnotové (0/1, nebo true/false), ale mohou také obsahovat celé soubory atd. Příkladem nám může být jakýkoliv internetový obchod, kde si budeme evidovat jednotlivé zákazníky a jejich nákupy. U nepřihlášených uživatelů si zaznamenáváme posloupnost jimi zobrazovaných jednotlivých stránek a nabízených produktů.

Dolování dat slouží pro jejich analyzování, kdy se snaží nacházet v rozsáhlých datech jednotlivé vzory, předpovídání trendů, vytváření doporučení a pravidel, popřípadě analyzovat sledy událostí. K hledání vzorů, trendů a předpovědí se využívají tyto metody:

- matematické,
- statistické,
- umělá inteligence.

Protože je dolování dat velice důležitým nástrojem, nabízejí ho dnes jako součást svých databázových systémů téměř všechny velké softwarové firmy (Microsoft, Oracle, DB2 atd.). V této práci jsme se zaměřili na databázový systém Microsoft SQL Server 2012 Standard Edition.

2.1 Proces dolování dat

Pro úspěšné vytváření modelů pro dolování dat musíme nejprve provést proces dolování dat. Jedná se o širší proces, který můžeme rozdělit do šesti částí:

1. definice problému,
2. příprava dat,
3. zkoumání dat,
4. vytváření modelů pro dolování dat,
5. prozkoumání a ověření modelu,
6. nasazení a aktualizace modelu.

Celý proces či jednotlivé kroky se mohou iterovat v rámci zlepšování výsledného modelu pro dolování dat. Jedná se tedy o dynamický a opakující se proces, kdy například při zkoumání dat

zjistíme, že nemáme dostatečné údaje pro úspěšné vytvoření modelu pro dolování dat. Musíme tedy tento krok opakovat a získat tak další potřebné údaje.

2.1.1 Definice problému

Na úplném začátku vytváření modelů pro dolování dat musíme znát strukturu vstupních dat, například zda se jedná o sekvenční data či nikoliv. Na první pohled nám nemusí dávat vstupní data smysl a my tak nepoznáme, co který záznam či atribut znamená. Jinak řečeno, jestliže máme vstupní data přímo v tabulkách databáze, pak u těchto tabulek potřebujeme znát jejich strukturu a propojení. Dalším důležitým faktorem je co chceme z těchto informací získat neboli vydolovat a co je tedy naším cílem.

2.1.2 Příprava dat

Zadaná vstupní data můžeme mít uložena v jakékoliv formě. V lepším případě obdržíme vstupní data v jakékoliv elektronické formě, například datové soubory různých formátů, či data v databázových tabulkách. Nicméně v horším případě můžeme obdržet vstupní data v neelektronické formě, například jako kartotéky. Jakmile vstupní data nemáme v databázových tabulkách ale v jakékoliv jiné formě, musíme provést přípravu obdržených dat. Následně provedeme možným způsobem import těchto dat do databázových tabulek.

2.1.3 Zkoumání dat

Vstupní data musíme chápat, abychom v nich při dolování byli schopni dosahovat správných výsledků. Jakmile je známe a chápeme, dokážeme rozhodnout, zda neobsahují chybné údaje. Příkladem může být, to že o určitém atributu víme, že dosahuje pouze hodnot nula a jedna. V takovémto případě se u daného atributu nesmí objevit žádná jiná hodnota.

2.1.4 Vytváření modelů pro dolování dat

V tomto kroku vytváříme výsledné modely pro dolování dat. Zde definujeme, jaký algoritmus chceme použít pro dolování dat. Dále si volíme vstupní sloupce, respektive sloupce, u nichž chceme předpovídat výsledky. V tomto kroku můžeme nastavovat a upravovat parametry vybraného algoritmu.

U vytvářených modelů se vstupní data rozdělují do dvou množin, a to do trénovací a testovací množiny. Poměr rozdělení vstupních dat do jednotlivých množin můžeme nastavovat pomocí parametrů.

Trénování modelu se provádí nad daty, které obsahuje trénovací množina. Z této množiny se model učí, tedy hledá vztahy, vazby, trendy atd., pomocí nichž pak vytváří doporučení, predikce apod. Při vytváření modelu zadáváme v procentech, kolik dat ze vstupní tabulky má patřit do této množiny.

Testovací množina se používá pro otestování naučeného modelu pro dolování v datech. Při vytváření modelu můžeme nastavit, kolik dat se bude nacházet v této testovací množině. Tento údaj zadáváme v procentech nebo maximálním počtem dat, která se mohou použít.

2.1.5 Prozkoumávání a ověření modelu

Nyní můžeme prozkoumat výsledný vytvořený model pro dolování dat. Žádoucím krokem je sestavení několika výsledných modelů pro dolování dat. Pro každý vytvořený model použijeme jiné nastavení parametrů. Výsledky modelů porovnáme a ověříme tak, který z modelů je nejpřesnější.

Často budeme odhadovat výsledky několika atributů v jednom modelu pro dolování dat. Proto je nezbytně nutné u každého vytvořeného modelu pro dolování dat ověřit přesnost odhadovaného atributu zvlášť. Pro každý atribut určíme, který z vytvořených modelů pro dolování dat je přesnější. Může nastat situace, že jeden model pro dolování dat bude podávat lepší předpovědi pro jeden konkrétní atribut a v druhém vytvořeném modelu pro druhý odhadovaný atribut. Proto je tedy dobré používat oba modely pro dolování dat zvlášť na každý odhadovaný atribut.

2.1.6 Nasazení a aktualizace modelů

Ověřený model můžeme začít používat pro odhadování trendů, předpovídání výsledků atd. Jestliže jsou vstupní data neustále aktualizována anebo doplňována, je žádoucí abychom vytvořený model pro dolování dat obnovovali. Pravidelným sestavováním se model bude vyvíjet v čase tak, jako se mění trendy, výsledky atp.

Příkladem nám může být odhadování dalších položek, které se mohou objevit v nákupním košíku. Pokud bychom využívali neaktualizovaný model pro dolování dat, který byl vytvořen již dříve, pak se v tomto modelu nemusí nacházet položky, které jsou momentálně nejvíce prodávány.

2.2 Použití dolování dat

Pomocí dolování dat můžeme principiálně studovat, pochopit a případně i vylepšit prakticky jakýkoliv proces ve vzájemně velmi odlišných oblastech, jako je například řízení procesu výroby, lidské zdroje, analýza lékařských vzorků, analýza signálů, prostě všude tam, kde je možné shromažďovat data z procesů [1]. Se správným marketingovým či analytickým myšlením, nám dolování dat může sloužit jako podpora při rozhodování.

Příkladem nám může být opět znovu nějaký internetový obchod, kdy si evidujeme zákazníky a jejich nákupy. Z těchto dat můžeme zjistit mnohé prospěšné informace. Například, které prodávané zboží spolu nejvíce souvisí. Při přidání nějakého kusu zboží do košíku můžeme ihned zákazníkovi nabídnout, zda nemá zájem o nějaké další zboží. Tuto nápovědu můžeme zobrazit formou nerušivých postranních našeptávačů či reklam. Typickým příkladem může být nákup jakékoliv tiskárny, kdy ihned zákazníkovi můžeme nabídnout dokoupení datového kabelu pro propojení tiskárny s počítačem, jelikož ve většině případů není standardem, aby tento kabel

byl součástí prodejního balení tiskárny. Dalším typickým nabízeným zbožím může být nabídka kancelářských papírů či fotopapírů, popřípadě náplní pro tisk a jiného spotřebního zboží souvisejícího s nákupem tiskárny.

2.3 K čemu dolování dat nelze použít

Vytvářený model pro dolování dat nelze použít, jestliže jsou vstupní data náhodně vygenerována. Z takovýchto informací nevydolujeme žádné použitelné, respektive reálné informace. Dalším případem může být, že ve vstupních datech nejsou veškeré informace, které bychom chtěli zkoumat a vydolovat z nich nějaké další informace. Příkladem nám může být závislost prodeje zimního zboží na teplotě vzduchu, kdy se při prodeji tohoto zboží neeviduje, jaká je aktuální venkovní teplota. Rovněž neznáme bydliště zákazníka a venkovní teploty v okolí jeho bydliště.

Výsledky, které nám dodávají modely pro dolování dat, nejsou stoprocentně přesné. Sloužit nám mohou pouze pro podporu rozhodování, nebo odhadování trendů, vytváření shluků atd. Dolování dat není všemocný nástroj, proto tedy nesmíme na výsledné modely plně spoléhat, ale slouží nám pouze jako informativní nástroj, či podpora při rozhodování.

2.4 Používané algoritmy v Microsoft SQL server 2012

Microsoft SQL Server 2012 obsahuje následující algoritmy pro vytváření modelů pro dolování dat:

- Microsoft Association Rules Algorithm
- Microsoft Clustering Algorithm
- Microsoft Decision Trees Algorithm
- Microsoft Linear Regression Algorithm
- Microsoft Logistic Regression Algorithm
- Microsoft Naive Bayes Algorithm
- Microsoft Neural Network Algorithm
- Microsoft Sequence Clustering Algorithm
- Microsoft Time Series Algorithm

V dalších kapitolách se naučíme vybrané algoritmy, jak pracují a pomocí kterých parametrů můžeme ovlivnit jednotlivé výsledné modely. Dále si ukážeme praktické příklady jak vytvářet modely pro dolování dat a získávat informace z těchto vytvořených modelů.

2.5 Výběr správného algoritmu

Výběr nejlepšího algoritmu pro jednotlivé typy úloh není vždy nejlehčí, protože každý je zaměřen na jiné vlastnosti, tedy používá odlišné metody pro dolování. Můžeme použít různé algoritmy na stejná vstupní data, ale každý algoritmus bude generovat jiné výsledky a některé mohou vytvářet i více výsledků.

2.5.1 Algoritmy dle typu dat

Microsoft SQL Server 2012 Analysis Services obsahuje následující typy algoritmů [2]:

- **Klasifikační algoritmy**: předpovídání jedné nebo více diskrétních proměnných na základě jiných atributů v datovém souboru.
- **Regresní algoritmy:** předpovídání jedné nebo více spojitých proměnných, jako je zisk nebo ztráta, které jsou založeny na jiných atributech v datovém souboru.
- **Segmentační algoritmy:** rozdělení dat do skupin nebo clusterů, které mají podobné vlastnosti.
- **Sdružovací algoritmy:** nalezení korelací mezi různými atributy v datové množině. Nejčastěji se aplikují pro vytváření asociačních pravidel, která mohou být použita v analýze nákupního košíku.
- **Sekvenční algoritmy:** shlukování častých sekvencí v datech, jako je průchod webovými stránkami.

Při vytváření modelu pro dolování dat jich můžeme vytvořit více. Proto zde není důvod, abychom byli omezeni pouze na jeden algoritmus pro dolování v datech. Můžeme tedy používat více algoritmů v jednom modelu. Jakmile máme v jednom modelu použito více algoritmů pro dolování dat, můžeme jednotlivé algoritmy používat pro předpovídání jednotlivých výsledků či atributů, kdy smíme použít vždy ten nejlepší algoritmus pro předpovídání konkrétního atributu.

2.5.2 Výběr algoritmu dle typu úlohy

Následující tabulka 1 obsahuje doporučení pro jednotlivé typy úkolů, pro které se tradičně používají jednotlivé algoritmy.

Tabulka 1: Výběr algoritmu dle typu úlohy [2]

Příklad úlohy	Algoritmy k použití
Předvídání diskrétních atributů: • Členění zákazníků na několik potencionálních kupujících jako dobré a špatné • Vypočítání pravděpodobnosti, že server selže během následujících 6 měsíců • Kategorizace výsledků pacienta a prozkoumání souvisejících faktorů	Microsoft Decision Trees Algorithm Microsoft Naive Bayes Algorithm Microsoft Clustering Algorithm Microsoft Neural Network Algorithm
Předvídání spojitých atributů: • Předpověď tržby pro příští rok • Předpovídání sezónních trendů na základě minulých let • Generování výsledku rizika dané demografie	Microsoft Decision Trees Algorithm Microsoft Time Series Algorithm Microsoft Linear Regresion Algorithm
Předvídání sekvencí: • Analýza prohlížení webových stránek • Analýza faktorů vedoucí k selhání serveru • Zachycení a analýza posloupností aktivit během ambulantních návštěv pacienta, vytvoření nejlepšího postupu při vyšetření	Microsoft Sequence Clustering Algorithm
Nalezení skupin společných položek v transakcích: • Analýza nákupního košíku a určení umístění produktu • Navrhování dalších produktů ke koupi • Průzkum a analýza zboží v akci, zjištění, která byla korelována, plánování budoucí činnosti	Microsoft Association Algorithm Microsoft Decision Trees Algorithm
Nalezení skupin s podobnými položkami: • Vytvoření rizikových skupin a profilů pacienta na základě demografie a příznaků • Analýza uživatelů procházení a nákupové vzory • Identifikování serverů, které mají společné využití	Microsoft Clustering Algorithm Microsoft Sequence Clustering Algorithm

3. Popis vybraných algoritmů

V této kapitole se budeme zabývat vybranými algoritmy, které jsou obsaženy v Microsoft SQL Server 2012 Analysis Services pro dolování dat. Popíšeme si vybrané algoritmy, na jakých principech pracují. Ukážeme si, které parametry obsahují a co pomocí těchto parametrů můžeme nastavit respektive ovlivnit. Všechny algoritmy pro dolování dat mají tyto parametry, které již nebudeme popisovat zvlášť u všech algoritmů:

- **HoldoutMaxCases:**
  Nastavením tohoto parametru se definuje maximální počet objektů, které se zahrnou v testovací sadě, jedná se o testovací množinu dat. Udává se celočíselná hodnota.

- **HoldoutMaxPercent:**
  Nastavením tohoto parametru se definuje počet dat, která chceme zahrnout do testovacího modelu jako procentní podíl z úplného souboru vstupních dat. Udává se v procentech v rozmezí 0 – 99. Jestliže nechceme mít žádnou testovací sadu, nastavíme na hodnotu 0.

- **HoldoutSeed:**
  Nastavením tohoto parametru definujeme, že se jednotlivé oddíly mohou opakovat. Nastavujeme v celočíselných hodnotách.

Každý algoritmus pro dolování dat podporuje pouze některé typy obsahu vstupních a předvídaných atributů. O jaké typy se jedná, je vždy uvedeno v požadavcích konkrétního algoritmu. Atribut je sloupec v tabulce, která je uložená v databázi. Při vytváření modelu pro dolování dat se typ obsahu sloupce nastavuje pro všechny atributy, které tabulka obsahuje. Pokud máme v tabulce atribut, ve kterém se opakují data v určitém intervalu, můžeme tento atribut nastavit na cyklický typ. K dispozici máme tyto typy [2]:

- **Discrete (diskrétní):**
  Atribut obsahuje konečný počet nespojitých hodnot. Jedná se například o atribut, ve kterém se zaznamenává věk osoby.

- **Continuous (spojité):**
  Atribut obsahuje spojité hodnoty v nějakém číselném rozsahu. Jedná se například o atribut, ve kterém se zaznamenává teplota.

- **Discretized (diskretizované):**
  Diskretizace je proces, kdy se spojité hodnoty převádějí na segmenty tak, že pak existuje omezený počet možných hodnot. Diskretizovat se mohou pouze číselná data. Jedná se o atribut, ve kterém jsou spojité hodnoty.

- **Key (klíč):**
  Atribut obsahuje jedinečný identifikátor celého řádku respektive objektu v tabulce.

- **Key Sequence (klíč sekvence):**
  Atribut obsahuje klíč sekvence, tedy sled nějakých událostí. Hodnoty jsou uspořádané, avšak nemusí mít stejnou vzdálenost. Jednotlivé sekvence tedy mohou být různě dlouhé.

- **Key Time (klíč času):**
  Atribut obsahuje hodnoty, které představuje časové měřítko.

- **Table (tabulka):**
  Atribut obsahuje další datové tabulky, s jedním nebo více atributy, které mohou mít jeden či více řádků. Může se jednat například o vnořené tabulky.

- **Cyclical (cyklické):**
  Atribut obsahuje pouze hodnoty, které jsou v cyklicky uspořádané množině. Jedná se například o dny v týdnu.

- **Ordered (uspořádané):**
  Atribut obsahuje hodnoty, ve kterých se nachází sekvence či jiné posloupnosti.

3.1 Microsoft Association Rules Algorithm

V češtině můžeme algoritmus Microsoft Association Rules Algorithm (dále jen MARA) označit jako algoritmus asociačních pravidel, jedná se o asociační algoritmus. Tento algoritmus je zaměřen na odkrývání různých vztahů v datech a souvislostmi mezi různými vztahy. Asociační pravidla se využívají například k analýze nákupního košíku, kdy na základě již nakoupených nebo přidaných položek do nákupního košíku analyzuje tento košík. Pomocí asociačních pravidel pak napovídá zákazníkovi, o které další zboží má projevit zájem. Skupina položek se nazývá itemset.

Asociační model se skládá z řady množin a položek pravidel, tyto množiny popisují, jak jsou v rámci případů jednotlivé položky seskupeny. Pravidla, která nalezne tento algoritmus, mohou být následně použita k vytváření předpovědi, která může vést k dalšímu nákupu položky, na základě položek, jež má již zákazník v nákupním košíku. MARA může mít potencionálně mnoho pravidel v rámci datové sady. Asociační pravidla používají dva parametry, podporu a pravděpodobnost. Těmito parametry se popisují skupiny položek a pravidel, která je generují.

Tento algoritmus společně s Microsoft Decision Trees Algorithm může být použit k analýze asociačních pravidel. Objevená pravidla, jež naleznou jednotlivé algoritmy v datech, se mohou lišit.

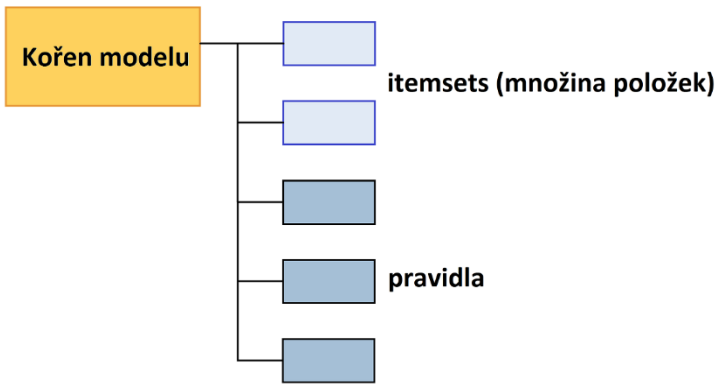
3.1.1 Jak pracuje algoritmus

MARA implementuje jednoduchý a známý Apriory algoritmus [3], vytváří a počítá kandidátské skupiny položek, neboli itemsets (množina položek). Jednotlivé položky mohou představovat produkty, události, nebo hodnotu atributů v závislosti na typu dat [2]. Jedná se o účinnou metodu pro dolování znalostí ve formě asociačních pravidel, probíhající ve dvou fázích. Nejprve jsou vyhledávány frekventované množiny položek (tj. ty, které dosahují minimální stanovené hodnoty

podpory) a z nich jsou generovány silné asociace (takové, které navíc dosahují i minimální určitosti) [4].

Pro každou skupinu položek algoritmus vypočítá skóre, které představuje podporu a důvěru. Na základě těchto výsledků pak můžeme získat zajímavá pravidla ze skupiny položek, tedy z itemsets.

Vytvořený asociační model má jednoduchou strukturu. Jednotlivé modely mají pouze jeden nadřazený uzel, který představuje model a jeho metadata. Každý rodič uzlu má seznam jednotlivých itemsetů a pravidel. Jednotlivé itemsety a pravidla se nijak neorganizují ve stromech. Model nejprve obsahuje jednotlivé itemsety a následně pravidla, znázorněno na obrázku 1.



Obrázek 1: Struktura asociačního modelu [2]

3.1.2 Požadavky

Pro úspěšné vytvoření modelu pro dolování dat pomocí algoritmu MARA musí vstupní data obsahovat klíčový sloupec, vstupní sloupce a jeden odhadovaný sloupec, přičemž podporuje pouze vstupní a předvídané sloupce následujícího typu:

- **Vstupní atributy:**
  Cyklické, diskrétní, diskretizované, klíčový sloupec, tabulka, uspořádané.

- **Předvídatelné atributy:**
  Cyklické, diskrétní, diskretizované, tabulka, uspořádané.

Cyklické a uspořádané typy jsou podporovány, ale algoritmus je zpracuje jako samostatné hodnoty.

3.1.3 Přizpůsobení

MARA umožňuje upravovat výsledný model pomocí parametrů, které ovlivňují chování, výkon a přesnost výsledného modelu. Parametry jsou při vytváření modelu ve výchozím stavu, kdy každý parametr je nastaven ve výchozí hodnotě. Pomocí následujících parametrů můžeme měnit výsledný model:

- **MAXIMUM_ITEMSET_COUNT**
  Nastavení maximálního počtu vytvořených itemsetů. Při vytváření modelu je nastavena výchozí hodnota 200 000.

- **MAXIMUM_ITEMSET_SIZE**
  Nastavení maximálního počtu položek v jednom itemsetu. Při vytváření modelu je nastavena výchozí hodnota 3. Jestliže nastavíme hodnotu na 0, počet položek v jednom itemsetu není nijak omezen.

- **MAXIMUM_SUPPORT**
  Nastavení maximálního počtu případů pro podporu, které itemset používá. Podpora (jinak řečeno frekvence) je počet případů, které obsahují jednotlivé položky nebo kombinace těchto položek. Slouží k odstranění prvků, jež se nám často opakují. Prvky, které se nám často opakují, mají malý význam. Při vytváření modelu je nastavena výchozí hodnota 1. Když je hodnota menší než 1, pak tato hodnota představuje jedno procento z celkového počtu případů. Jestliže je hodnota větší než 1, tak tato hodnota představuje absolutní počet případů, které může obsahovat itemset.

- **MINIMUM_ITEMSET_SIZE**
  Nastavení minimálního počtu položek, které jsou povoleny v itemset. Při vytváření modelu je nastavena výchozí hodnota 1. Jakmile tento počet zvýšíme, vytvořený model může celkově obsahovat méně itemsetů. Tento parametr slouží k ignoraci itemsetů. To nám může být užitečné, když chceme ignorovat itemsety, kde se nachází méně položek, než je zadaná hodnota.

- **MINIMUM_PROBABILITY**
  Nastavení minimální pravděpodobnosti, že dané pravidlo je pravdivé. Při vytváření modelu je nastavena výchozí hodnota 0,4. Generuje pouze ta pravidla, která mají větší pravděpodobnost, než jaká je nastavena hodnota.

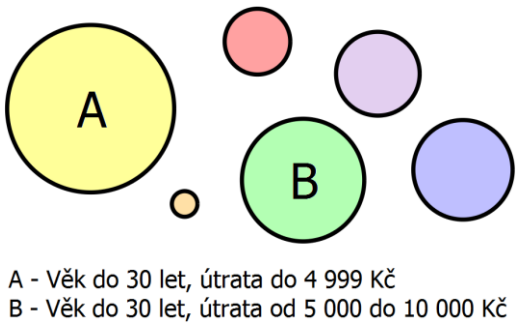
- **MINIMUM_SUPPORT**
  Nastavení minimálního počtu případů, které musí obsahovat každý vygenerovaný itemset. Při vytváření modelu je nastavena výchozí hodnota 0,03. Když tuto hodnotu nastavíme menší než 1, pak se minimální počet případů vypočítává jako procento z celkového počtu případů. Ale pokud nastavíme tuto hodnotu na celé číslo větší než 1, tak tím určujeme minimální počet případů, které musí obsahovat každý itemset. Jestliže není dostatek paměti, algoritmus může automaticky zvýšit hodnotu tohoto parametru.

- **OPTIMIZED_PREDICTION_COUNT**
  Nastavení počtu položek pro optimalizaci předpovědi, které mají být v paměti. Při vytváření modelu je nastavena výchozí hodnota 0. Když je použita výchozí hodnota tohoto parametru, algoritmus vytvoří tolik předpovědí, kolik jich existuje. Jakmile zadáme nenulovou hodnotu, tak predikce z dotazu vrátí maximálně zadaný počet položek. Nastavením tohoto parametru můžeme zvýšit výkon predikce. Například když je hodnota nastavena na 3, algoritmus uloží do paměti pouze 3 položky pro predikci. Další předpovědi se již nezobrazí, i když mohou mít stejnou pravděpodobnost jako tyto 3 zobrazené položky.

3.2 Microsoft Clustering Algorithm

Microsoft Clustering Algorithm (dále jen MCA) je určen pro shlukování dat. Tento algoritmus vytváří skupiny označované jako shluky nebo klastry. Do jednotlivých shluků dává objekty, které mají podobné vlastnosti. Pomocí těchto skupin můžeme přibližně určit vlastnosti objektů v jednotlivých skupinách. Model dokáže identifikovat vztahy v datové sadě, které my nemusíme logicky odvodit. Například může vypozorovat u jednotlivých zákazníků, jaké zboží nakupují a podle toho je rozdělit do jednotlivých skupin. Jednotlivé shluky se mohou, avšak nemusí překrývat. Pro ilustraci se můžeme podívat na obrázek 2.

Shlukovací algoritmus se liší od ostatních algoritmů pro dolování v datech, neboť k úspěšnému vytvoření výsledného modelu pro dolování dat nepotřebuje žádný odhadovaný sloupec. Je to z toho důvodu, protože MCA pouze shlukuje objekty se stejnými vlastnostmi. MCA trénuje vytvářený model pouze na vztazích, které existují v datech, kde tyto vztahy sám dokáže identifikovat.



Obrázek 2: Ukázka shluků

3.2.1 Jak pracuje algoritmus

Nejprve MCA identifikuje vztahy v datech, na kterých se vytváří tento model. Na základě vztahů, které našel v datech, si následně model generuje jednotlivé klastry. Po prvotním výpočtu klastrů a objektů v jednotlivých klastrech vypočítává, jak dobře obsadil jednotlivé objekty do klastrů, případně jednotlivé objekty přesune. Tento výpočet se opakuje tak dlouho, dokud nenalézá lepší výsledek.

MCA využívá pro vytváření klastrů dvě metody K-means a Expectation Maximation (dále jen EM). Metoda K-means je označována jako „těžké“ shlukování, to znamená, že každý

objekt může patřit pouze do jednoho klastru. Pro každý objekt, který je v klastru se vypočítává, s jakou pravděpodobností je jeho součástí. EM metoda je označování jako „měkká“ metoda shlukování, to znamená, že každý objekt může patřit do více shluků najednou. Proto se pro jeden objekt vypočítávají všechny kombinace pravděpodobností, zda je součástí daného klastru. Metodu, kterou chceme použít v modelu pro dolování dat, si volíme pomocí parametru.

K-means

Detailnější popis této metody nalezneme v dokumentu [5]. Zde se budeme zabývat teoretickým minimem pro základní přehled této metody.

K-means neboli k-středová metoda, jedná se o známou metodu pro vytváření shluků. Kdy při minimálních rozdílech mezi objekty zachovává maximální vzdálenosti mezi shluky. Počáteční objekty jsou zvoleny náhodně, tyto objekty vytvářejí střed (means) klastru neboli těžiště. Z těchto středů jsou iterativně přepočítány všechny objekty v klastru. Písmeno „K“ označuje libovolný počet shluků. Obvykle se zadává nějakým parametrem, v našem případě se jedná o parametr *cluster_count*, který si vysvětlíme níže.

Tato metoda počítá euklidovskou vzdálenost mezi jednotlivými objekty v klastru. V klastru neboli v těžišti dochází ke konvergenci mezi ostatními objekty a to až do té doby, než dosáhnou minimální hodnoty konvergence. Každý objekt je součástí pouze jednoho klastru. Důležitost objektu v klastru je dána vzdáleností od těžiště.

Obvykle se K-means metoda používá pro vytváření klastrů spojitých datových typů, kde se pro výpočet používají vzdálenosti ke K-středové hodnotě. Avšak MCA upravuje tuto metodu klastrů na diskrétní atributy pomocí pravděpodobnosti. Výpočet pro diskrétní atributy se provádí dvěma způsoby. Neškálovatelný (non-scalable) K-means, kde se načtou všechny objekty a udělá se pouze jeden průchod shlukování. Druhý způsob je škálovatelný (scalable) K-means, kdy algoritmus použije prvních 50 000 objektů, více objektů používá pouze v případě, kdy je potřeba více dat, aby se dosáhlo lepšího výsledku modelu. [2]

EM Clustering

Neboli Expectation Maximatization Clustering. Algoritmus vychází z principu K-means. Tento algoritmus umožňuje, aby objekt mohl současně patřit do více shluků najednou.

Obecný princip fungování EM algoritmu [6]:

1. Stanovení požadovaného počtu K shluků.
2. Náhodný výběr výchozích K jader.
3. E-krok: přiřazení pravděpodobnostní hodnoty příslušnosti ke shlukům.
4. M-krok: provedení nových odhadů parametrů s využitím právě vypočtených hodnot.
5. Opakování kroků 3. a 4. do té doby, kdy změny všech parametrů jsou menší, než je zvolená hranice.

Stejně jako K-means, tak tento algoritmus je implementován v MCA ve dvou variantách. První varianta je škálovatelný EM a druhý je neškálovatelný EM. Škálovatelný EM používá prvních 50 000 objektů pro počáteční sken. Jestliže nemůže úspěšně vytvořit model, načte se dalších 50 000 objektů. Neškálovatelný EM načte všechny objekty bez ohledu na jejich velikost. Požadavky na paměť jsou tím pádem o mnoho vyšší, ale zase vytvořený model je o to přesnější. [2]

3.2.2 Požadavky

Pro úspěšné vytvoření modelu pro dolování dat pomocí algoritmu MCA musí vstupní data obsahovat klíčový sloupec a vstupní sloupce, přičemž podporuje vstupní a předvídané sloupce následujícího typu:

- **Vstupní atributy:**
  Spojité, cyklické, diskrétní, diskretizované, klíčový sloupec, tabulka, uspořádané.

- **Předvídatelné atributy:**
  Spojité, cyklické, diskrétní, diskretizované, tabulka, uspořádané.

Cyklické a uspořádané typy jsou podporovány, ale algoritmus je zpracuje jako samostatné hodnoty.

3.2.3 Přizpůsobení

MCA umožňuje upravovat výsledný model pomocí parametrů, které ovlivňují chování, výkon a přesnost výsledného modelu. Parametry jsou při vytváření modelu ve výchozím stavu, kdy každý parametr je nastaven ve výchozí hodnotě. Pomocí následujících parametrů můžeme měnit výsledný model:

- **CLUSTERING_METHOD**
  Nastavením tohoto parametru určujeme, kterou metodu pro shlukování chceme použít. Při vytváření modelu je nastavená výchozí hodnota 1. Nastavit můžeme tyto hodnoty parametru, přičemž si vybereme metodu pro vytváření shluků:

  1. Škálovatelný EM.
  2. Neškálovatelný EM.
  3. Škálovatelný K-means.
  4. Neškálovatelný K-means.

- **CLUSTER_COUNT**
  Nastavením tohoto parametru určujeme přibližný počet klastrů, které mají být vytvořeny. Při vytváření modelu je nastavena výchozí hodnota 10. Jakmile nemůže být vytvořen model z přibližného počtu klastrů, MCA vytvoří tolik klastrů, kolik je možných. Když nastavíme hodnotu na 0, algoritmus sám odhadne nejlepší počet klastrů.

- **CLUSTER_SEED**
  Nastavením tohoto parametru určujeme, kolik objektů se v počáteční fázi použije k vytvoření klastrů. Při vytváření modelu je nastavena výchozí hodnota 0. Změnou tohoto čísla můžeme měnit počáteční klastry a následně porovnávat výsledné klastry. Jakmile se jednotlivé shluky již moc nemění, lze vytvořený model považovat za stabilní.

- **MINIMUM_SUPPORT**
  Nastavením toho parametru určujeme minimální počet objektů, které jsou nutné k vytvoření klastru. Při vytváření modelu je nastavena výchozí hodnota 1. Když je počet objektů menší než zvolené číslo, klastr se považuje za prázdný, a tím se odstraní.

- **MODELLING_CARDINALITY**
  Nastavením tohoto parametru určujeme počet vzorových modelů, které jsou vytvořeny během procesu vytváření shluků. Při vytváření modelu je nastavena výchozí hodnota 10. Snížením tohoto parametru můžeme zlepšit výkon algoritmu na úkor toho, že mohou chybět některé dobré kandidátní modely.

- **STOPPING_TOLERANCE**
  Nastavením tohoto parametru určujeme, kdy je dosažena konvergence, a tím algoritmus dokončil vytváření výsledného modelu. Při vytváření modelu je nastavena výchozí hodnota 10. Když je celková změna klastru pravděpodobnosti menší než poměr tohoto parametru rozděleného podle velikosti modelu, je dosaženo konvergence.

- **SAMPLE_SIZE**
  Nastavením tohoto parametru určujeme počet objektů, které algoritmus používá při průchodu. Při vytváření modelu je nastavena výchozí hodnota 50 000. Jakmile nastavíme na hodnotu 0, v jednom průchodu se prochází všechny vstupní objekty, ze kterých budou vytvořeny jednotlivé shluky.

- **MAXIMUM_INPUT_ATTRIBUTES**
  Nastavením tohoto parametru určujeme počet vstupních atributů daných objektů, které dokáže zpracovat dříve, než algoritmus vyvolá výběrovou funkci. Při vytváření modelu je nastavena výchozí hodnota 255. Jakmile nastavíme na hodnotu 0, říkáme tím, že neexistuje omezení pro počet použitých vstupních atributů.

- **MAXIMUM_STATES**
  Nastavením tohoto parametru určujeme maximální počet stavů u atributů, které algoritmus podporuje. Při vytváření modelu je nastavena výchozí hodnota 100. Když má atribut více stavů, než je stanovená hodnota, využívají se nejpoužívanější stavy do tohoto omezujícího počtu, zbylé stavy ignoruje.

3.3 Microsoft Decision Trees Algorithm

V češtině můžeme algoritmus Microsoft Decision Trees Algorithm (dále jen MDTA) označit jako algoritmus rozhodovacích stromů. Tento algoritmus se používá pro odhadování diskrétních a spojitých atributů. U tohoto algoritmu můžeme předpovídat výsledky pro více atributů najednou. Pro každý předvídaný sloupec (atribut) se vytváří samostatně jeden rozhodovací strom. Tedy jestliže máme více předvídaných atributů, tolik rozhodovacích stromů bude obsahovat výsledný model.

Používá se pro odhadování výsledků na základě již známých pravidel, tato pravidla nalezne při vytváření modelu. Příkladem nám může být zjištění, který zákazník se může stát znovu potencionálním zákazníkem. Dále můžeme určit pravděpodobnost, zda si zákazník koupí určitý výrobek například na základě věku.

Získané informace u tohoto algoritmu si ukládá, respektive vytváří rozhodovací strom. Jedná se o orientované grafy, které svým vzhledem připomínají strom [7]. Všechny varianty řešení se vizualizují pomocí uzlů, kde se pro jednotlivé uzly vypočítává pravděpodobnost.

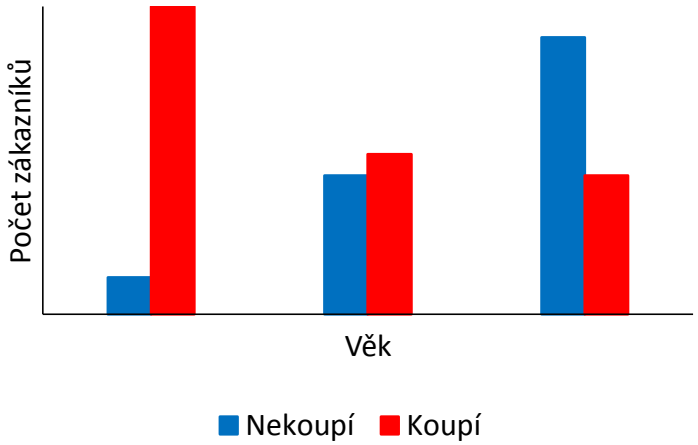
Rozhodovací strom [8] obsahuje uzly a větvení, jednotlivé větve pak představují jednotlivé varianty řešení. Každá větev rozhodovacího stromu je ohodnocena pravděpodobností. Kdybychom chtěli vypočítat pravděpodobnost nějaké cesty mezi dvěma uzly, museli bychom vynásobit všechny pravděpodobnosti na cestě mezi těmito dvěma uzly pomocí Bayesova vzorce. Tímto bychom získali pravděpodobnost, že se ze startovacího uzlu dostaneme do uzlu koncového. Bayesův vzorec plyne z podmíněné pravděpodobnosti [8]:

$$P(B_k|A) = \frac{P(A|B_k) * P(B_k)}{\sum_{i=1}^{n} P(A|B_i) * P(B_i)}$$

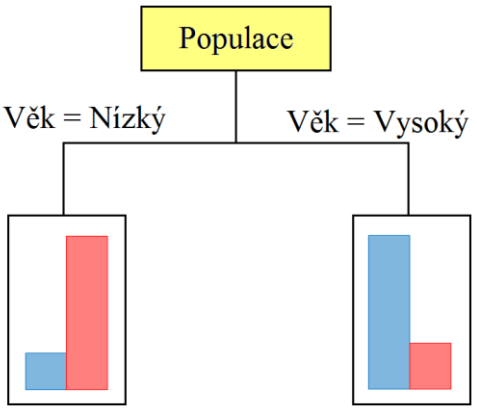
Pomocí Bayesova vzorce dokážeme vypočítat podmíněnou pravděpodobnost $P(B_k|A)$ jevu B_k za podmínky, že nastal jev A. Detailnější vysvětlení nalezneme v dokumentu [8].

Dále si tedy nebudeme popisovat, jak pracuje MDTA, detailnější popis nalezneme v dokumentaci [2]. Přidání nového uzlu do stromu se provádí tehdy, když algoritmus zjistí, že vstupní sloupec značně koreluje s předvídaným sloupcem. Jakým způsobem se rozdělí rozhodovací strom, závisí na tom, zda se jedná o spojitý či diskrétní sloupec.

Pro diskrétní data algoritmus vytváří předpovědi na základě vztahů mezi vstupními atributy objektů v datech. Při vytváření předpovědi konkrétního atributu používá pouze hodnoty, kterých nabývá předvídaný sloupec. Hodnoty těchto stavů se naučí při vytváření modelu. Algoritmus identifikuje korelace mezi vstupními sloupci a předvídaným sloupcem. Vytváření rozhodovacího stromu pro diskrétní data lze demonstrovat například pomocí histogramu, který se nachází na obrázku 3. Vytvoření nového uzlu můžeme vidět na obrázku 4, kdy najde korelaci pro vstupní atribut a předvídaný atribut.

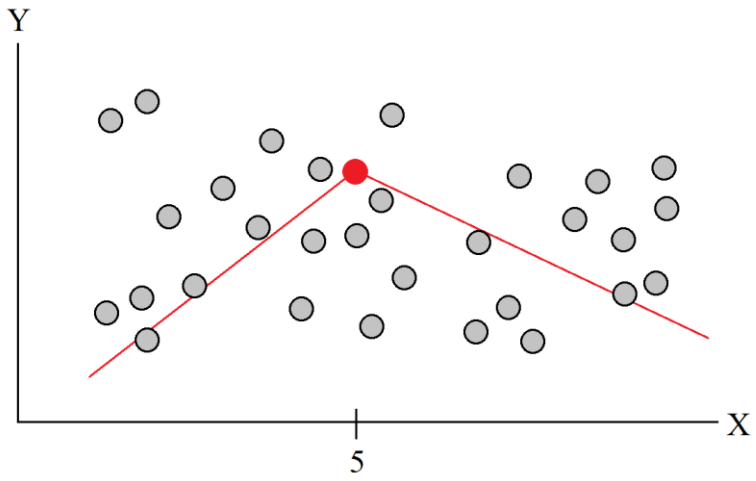


Obrázek 3: Ukázkový histogram závislosti koupě zboží na věku [2]

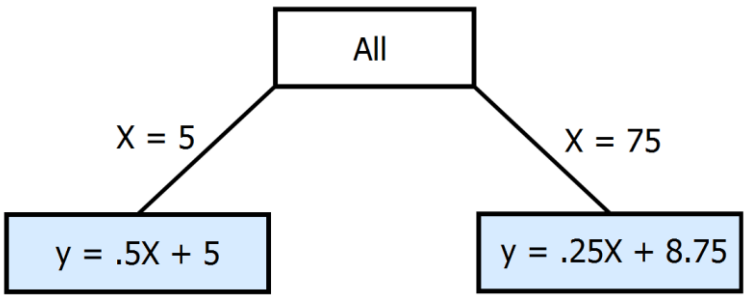


Obrázek 4: Ukázka vytvoření nového uzlu [2]

Pro spojité odhadované atributy algoritmus využívá lineární regrese k určení, kde se má rozhodovací strom rozdělit. Každý vytvořený uzel má svůj regresní vzorec. Rozdělení nastává v bodě zlomu, kde není lineární. Na obrázku 5 můžeme vidět takový bod zlomu nelinearity a poté na obrázku 6 vidíme vytvoření nového uzlu včetně regresního vzorce.



Obrázek 5: Ukázkový bod zlomu nelinearity [2]



Obrázek 6: Ukázka vytvořeného uzlu, včetně regresního vzorce [2]

3.3.1 Jak pracuje algoritmus

Na začátku vytváření modelu využívá algoritmus Bayesovu metodu k učení a získání přibližného rozdělení rozhodovacího stromu. Jedná se o statistickou metodu, která vypočítává podmíněnou pravděpodobnost, což již bylo popsáno výše. Při vytváření modelu pomocí algoritmu MDTA si nejprve algoritmus vytváří množiny vstupních hodnot. Následně identifikuje jednotlivé atributy a hodnoty objektů. Z těchto dat získává údaje, které obsahují nejvíce informací, respektive jsou klíčové v zkoumaných objektech. Ostatní hodnoty, které nejsou moc zastoupeny u jednotlivých atributů, odstraňuje. Rozhodovací strom vychází z korelace mezi vstupními a výslednými hodnotami. Jakmile byly nalezeny všechny korelované atributy, algoritmus nalezne jeden atribut, který čistě rozdělí jednotlivé výsledky. Takovýto atribut se měří pomocí rovnice rozdělení, která vypočítává zisk informace. Když je takovýto atribut nalezen, dojde k rozdělení rozhodovacího stromu a vytvoření nového uzlu. Tento postup se opakuje do té doby, dokud již nelze dále rozhodovací strom rozdělit. Rovnice, podle které se vypočítává zisk informace, závisí na parametrech algoritmu, datových vstupech atributů a typu předvídaného atributu.

Když máme předvídaný atribut a vstupní atributy diskrétního datového typu, výpočet výsledného modelu se provádí pomocí matice a následného vytvoření skóre pro každý bod

v matici. Pokud ale máme vstupní sloupce spojitého datového typu a předvídaný sloupec je diskrétního datového typu, pak jsou vstupní datové typy diskretizovány. Jestliže jsou všechny atributy spojitého datového typu, používá se metoda lineární regrese k vytvoření rozhodovacího stromu.

Pro výpočet důležitosti informace MDTA používá několik metod analýz, které jsou uvedeny v tabulce 2. Jejich detailnější popis nalezneme v [2].

Tabulka 2: Metody analýz pro výpočet informačního zisku informace [2]

Algoritmus	Metoda analýzy	Poznámky
Rozhodovací stromy	• Interestingness score • Shannon's Entropy • Bayesian with K2 Prior • Bayesian Direchlet with uniform prior	Jestliže některé sloupce obsahují jiné než nebinární spojité hodnoty, používá se Intersingness score, aby byla zajištěna konzistence dat.
Lineární regrese	• Interestingness score	Lineární regrese používá pouze Interestingnes score, protože podporuje spojité datové typy.

3.3.2 Požadavky

Pro úspěšné vytvoření modelu pro dolování dat pomocí algoritmu MDTA musí vstupní data obsahovat klíčový sloupec, vstupní sloupce a alespoň jeden odhadovaný sloupec, přičemž podporuje vstupní a předvídané sloupce následujícího typu:

- **Vstupní atributy:**
  Spojité, cyklické, diskrétní, diskretizované, klíčový sloupec, tabulka, uspořádané.

- **Předvídatelné atributy:**
  Spojité, cyklické, diskrétní, diskretizované, tabulka, uspořádané.

Cyklické a uspořádané typy jsou podporovány, ale algoritmus je zpracuje jako samostatné hodnoty.

3.3.3 Přizpůsobení

MDTA umožnuje upravovat výsledný model pomocí parametrů, ovlivňuje chování, výkon a přesnost výsledného modelu. Parametry jsou při vytváření modelu ve výchozím stavu, kdy každý parametr je nastaven ve výchozí hodnotě. Pomocí následujících parametrů můžeme měnit výsledný model:

- **COMPLEXITY_PENALTY**
  Nastavením tohoto parametru reguluje růst rozhodovacího stromu. Při vytváření modelu je nastavena výchozí hodnota podle počtu atributů takto:

- 1 až 9 atributů, je nastavena výchozí hodnota 0,5.
- 10 až 99 atributů, je nastavena výchozí hodnota 0,9.
- 100 a více atributů, je nastavena výchozí hodnota 0,99.

Vysoká hodnota tohoto parametru snižuje počet rozdělení ve stromu, naopak nízká hodnota tohoto parametru zvyšuje počet rozdělení ve stromu.

- **FORGE_REGRESSOR**
  Nastavením tohoto parametru přinutíme použít specifikované sloupce bez ohledu na to, které vypočítá sám algoritmus. Používá se pouze pro rozhodovací stromy, které předpovídají spojité datové atributy. Úprava tohoto parametrů je dostupná pouze v některých edicích Microsoft SQL Server.

- **MAXIMUM_INPUT_ATTRIBUTES**
  Nastavením tohoto parametru definujeme počet vstupních atributů, které se používají při vytváření rozhodovacího stromu. Filtrovací funkce atributů se provede tehdy, bude-li použito více vstupních atributů, než je povolená maximální hodnota. Při vytváření modelu je nastavena výchozí hodnota 255. Jakmile chceme používat všechny atributy a algoritmus nemá provádět filtraci atributů, nastavíme parametr na 0.

- **MAXIMUM_OUTPUT_ATTRIBUTES**
  Nastavením tohoto parametru definujeme počet výstupních atributů, které algoritmus dokáže zpracovat, než se provede filtrování atributů. Při vytváření modelu je nastavena výchozí hodnota 255. Jakmile chceme používat všechny atributy a algoritmus nemá provádět filtraci atributů, nastavíme parametr na 0.

- **MINIMUM_SUPPORT**
  Nastavením tohoto parametru definujeme minimální počet listů, které jsou potřebné pro vygenerování rozdělení rozhodovacího stromu. Při vytváření modelu je nastavena výchozí hodnota 10. Jestliže máme příliš mnoho vstupních dat, musíme tuto hodnotu zvýšit, neboť by mohlo dojít k přeučení rozhodovacího stromu.

- **SCORE_METHOD**
  Nastavením tohoto parametru určujeme metodu, která se používá pro výpočet rozdělovacího skóre. Metody, které můžeme nastavit, včetně výchozí metody, vidíme v tabulce 3.

Tabulka 3: Metody rozdělovacího skóre [2]

ID	Metoda
1	Entropy
3	Bayesian with K2 Prior
4 (výchozí hodnota)	Bayesian Direchlet Equivalent with uniform prior

- **SPLIT_METHOD**
  Nastavením tohoto parametru určujeme metodu, která se použije při rozdělování uzlů. Metody, které můžeme nastavit, včetně výchozí metody vidíme v tabulce 4.

Tabulka 4: Metody rozdělování uzlů [2]

ID	Metoda
1	**Binární** - strom je rozdělen do dvou větví, bez ohledu na skutečný počet hodnot atributu.
2	**Úplná** - strom může vytvořit tolik rozdělení, kolik je hodnot atributů.
3 (výchozí hodnota)	**Obě** - algoritmus sám určí, zda použije binární nebo úplné rozdělení pro dosažení lepších výsledků.

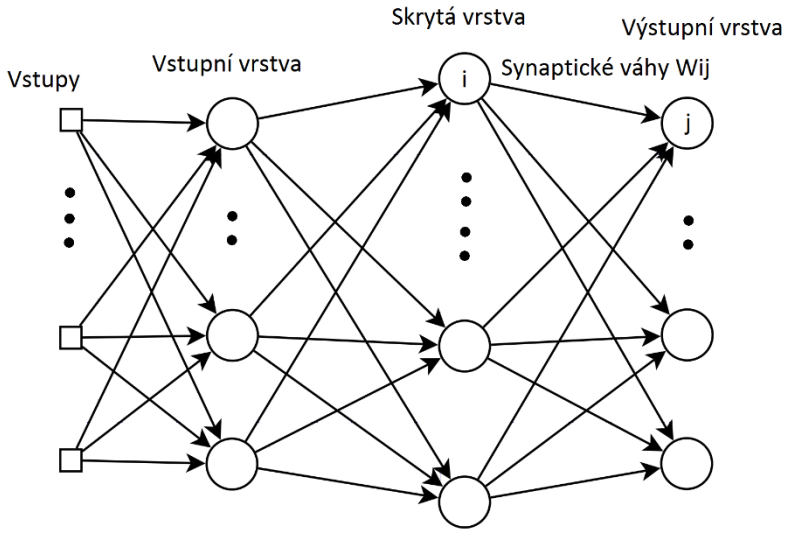
3.4 Microsoft Neural Network Algorithm

V češtině můžeme algoritmus Microsoft Neural Network Algorithm (dále jen MNNA) označit jako algoritmus neuronových sítí. MNNA si vytváří množiny, kdy spojuje každý vstupní atribut s každým odhadovaným stavem atributu. Pro výpočet pravděpodobnosti využívá trénovací data. Jedná se o regresní algoritmus, kdy na základě pravděpodobnosti může předpovídat odhadovaný atribut pomocí vstupních atributů. Výsledný model pro dolování dat pomocí MNNA může obsahovat více neuronových sítí. Počet těchto vytvořených sítí závisí na závislosti počtu vstupních sloupců, které se používají jako vstupní a odhadované atributy, ale také na tom, které atributy jsou odhadovány.

Používá se pro odhadování komplexních dat, u nichž máme velké množství vstupních dat. Pokud rozhodovací pravidla nelze snadno odvodit pomocí jiných algoritmů pro dolování dat, které jsou obsaženy v Microsoft SQL Server 2012, doporučuje se použít algoritmus MNNA. Může se jednat o různé výrobní procesy, obchodní postupy atd. Tento algoritmus se doporučuje pro následující scénáře [2]:

- Marketing a propagační analýzy, měření úspěšnosti propagace jako je rozesílání reklamních emailů, reklama v rádiích apod.
- Předvídání nutných rozhodnutí, kolísání měny, nebo jiné peněžní toky na základě dat z předešlých let.
- Analýza výrobních a průmyslových procesů.
- Dolování textu.
- Jakýkoliv odhadový model, který analyzuje složité vztahy mezi mnoha vstupy a poměrně málo výstupy.

Vytvořená neuronová síť se skládá až ze tři vrstev neuronů. Tyto vrstvy se rozdělují do vstupní vrstvy, volitelně skryté vrstvy a jako poslední je výstupní vrstva. Na obrázku 7, můžeme vidět, jak vypadá vícevrstvá neuronová síť.



Obrázek 7: Vícevrstvá neuronová síť

3.4.1 Jak pracuje algoritmus

MNNA využívá vícevrstvou perceptronovou síť, kde využívá algoritmus zpětného šíření (back-propagation), která je sestavována až ze tří vrstev neuronů neboli perceptronů. Perceptron [9] [10] je jedním z nejdůležitějších modelů. Je definován jako vážený součet vstupujících signálů. Pro hlubší poznání na jakém principu pracuje výpočet vah perceptronů, nám mohou posloužit tyto zdroje [9] [10]. Jednotlivé vrstvy se dělí na vstupní, skrytou a výstupní vrstvu. Každý neuron má jeden nebo i více vstupů, přičemž každý neuron vytváří jeden nebo více výstupů, viz obrázek 7.

Vstupní vrstva zpracovává a poskytuje vstupní hodnoty jednotlivých atributů pro dolování dat, které definují všechny hodnoty vstupních atributů pro vytvoření modelu a jejich pravděpodobností. Vstupní neurony zpracovávají jak diskrétní, tak spojité hodnoty vstupního atributu. Jeden vstupní neuron představuje jednu hodnotu vstupního atributu. Pokud má jeden atribut více diskrétních hodnot, vytvoří se tolik vstupních neuronů, kolik je diskrétních hodnot. Jestliže algoritmus nalezne prázdné hodnoty u vstupního atributu, vytvoří se další neuron pro chybějící diskrétní vstupní hodnoty. Pro spojité hodnoty vstupního atributu se vytvářejí dva neurony, a to pro chybějící stav či stavy a další pro spojitou hodnotu vstupního atributu. Jeden vstupní neuron může být vstupem pro jeden a více skrytých neuronů.

Skrytá vrstva obsahuje skryté neurony, které získávají hodnoty pravděpodobností od vstupních neuronů. Ve skryté vrstvě se z těchto hodnot vypočítává váha jednotlivých neuronů. Každý neuron je tedy ohodnocen váhou, která určuje jeho význam nebo význam daného vstupu do skrytého neuronu, což odpovídá stavu excitace [9] (vybuzení, zesílení). Vypočtená váha může dosahovat negativních hodnot, což odpovídá stavu inhibice [9] (utlumení, omezení). Hodnoty, které získala skrytá vrstva od vstupní vrstvy, poskytuje dál výstupní vrstvě.

Výstupní vrstva obsahuje odhadované výsledky jednotlivých atributů. Počet výstupních neuronů závisí na počtu předvídaných hodnot odhadovaného atributu, včetně chybějící hodnoty. Když odhadujeme výsledek atributu datového typu boolean, počet výstupních neuronů bude tři. Jeden pro hodnotu pravda, druhý pro hodnotu nepravda a poslední pro neexistující hodnotu. U dvou diskrétních hodnot se vygeneruje jeden neuron pro každý stav, plus jeden další neuron pro chybějící či neexistující odhadovanou hodnotu. Pro spojité hodnoty odhadovaného atributu se vytvoří dva výstupní neurony. Jeden neuron pro spojitou hodnotu odhadovaného sloupce a druhý pro chybějící či neexistující hodnotu. Když se při vytváření modelu pro dolování dat vygeneruje více než 500 výstupních neuronů pro odhadovaný atribut, tak si algoritmus vytvoří další novou neuronovou síť.

Každý výstupní a skrytý neuron má ohodnocen své vstupy váhovou hodnotou, která udává význam daného vstupu. Čím větší váhu neuron má u vstupu, tím je hodnota tohoto vstupu důležitější. Naopak může váha jednotlivých neuronů nabývat i negativních hodnot, tím pádem mohou být jednotlivé vstupy diskriminovány.

Každý neuron má přiřazenou jednoduchou lineární funkci, nazývanou jako aktivační funkce, která popisuje význam určitého neuronu k dané vrstvě neuronové sítě. Skryté neurony používají funkci hyperbolic tangent function (tanh) [2] pro jejich aktivaci, kdežto výstupní neurony používají pro aktivaci sigmoid function [2]. Obě funkce jsou nelineární a spojité. Tyto funkce umožnují neuronovým sítím modelovat nelineární vztahy mezi vstupními a výstupními neurony.

Při učení neuronové sítě si algoritmus nejprve testováním vyhodnotí trénovací data, která slouží při posuzování přesnosti sítě. Během trénování modelu je po každé iteraci neustále vyhodnocována přesnost neuronové sítě pomocí trénovacích dat. Jakmile se již přesnost nezvyšuje, je učící fáze tohoto modelu ukončena.

3.4.2 Požadavky

Pro úspěšné vytvoření modelu pro dolování dat pomocí algoritmu MNNA, musí vstupní data obsahovat alespoň jeden vstupní a jeden výstupní sloupec, přičemž podporuje vstupní a předvídané sloupce následujícího typu:

- **Vstupní atributy:**
  Spojité, cyklické, diskrétní, diskretizované, klíčový sloupec, tabulka, uspořádané.

- **Předvídatelné atributy:**
  Spojité, cyklické, diskrétní, diskretizované, tabulka, uspořádané.

Cyklické a uspořádané typy jsou podporovány, ale algoritmus je zpracuje jako samostatné hodnoty.

3.4.3 Přizpůsobení

MNNA umožnuje upravovat výsledný model pomocí parametrů, které ovlivňují chování, výkon, a přesnost výsledného modelu. Parametry jsou při vytváření modelu ve výchozím stavu, kdy

každý parametr je nastaven ve výchozí hodnotě. Pomocí následujících parametrů můžeme měnit výsledný model:

- **HIDDEN_NODE_RATIO**
  Nastavením tohoto parametru definujeme poměr skrytých neuronů na vstupních a výstupních neuronech. Při vytváření modelu je nastavena výchozí hodnota 4,0. Vzorec pro výpočet počátečních neuronů ve skryté vrstvě vypadá takto:

  $$HIDDEN\_NODE\_RATIO * \sqrt{a * b}$$

  Znak *a* označuje celkový počet vstupních neuronů. Znak *b* označuje celkový počet výstupních neuronů.

- **HOLDOUT_PERCENTAGE**
  Nastavením tohoto parametru definujeme ukončovací kritérium při učení výsledného modelu. Udává se jako procento ze vstupních dat pro školení k výpočtu chyby. Při vytváření modelu je nastavena výchozí hodnota 30.

- **HOLDOUT_SEED**
  Nastavením tohoto parametru určujeme, kolik objektů se použije v počáteční fázi. Výběr dat, která se použijí k učení modelu, zajišťuje náhodný generátor pro výběr dat. Nastavením parametru na hodnotu 0 se výběr dat řídí algoritmem MNNA. To nám zaručuje, že v průběhu zpracovávání zůstává obsah modelu vždy stejný. Při vytváření modelu je nastavena výchozí hodnota 0.

- **MAXIMUM_INPUT_ATTRIBUTES**
  Nastavením tohoto parametru definujeme počet vstupních atributů, které se používají při vytváření neuronové sítě. Pokud počet vstupních atributů dosáhne maximálního počtu, musí se provést filtrovací funkce, která vybere atributy, jež se budou používat. Při vytváření modelu je nastavena výchozí hodnota 255. Pokud chceme používat všechny vstupní atributy a algoritmus nemá provádět filtraci atributů, nastavíme parametr na 0.

- **MAXIMUM_OUTPUT_ATTRIBUTES**
  Nastavením tohoto parametru definujeme počet výstupních atributů, které se používají při vytváření neuronové sítě. Pokud počet výstupních atributů dosáhne maximálního počtu, musí se provést filtrovací funkce, která vybere atributy, které se budou používat. Při vytváření modelu je nastavena výchozí hodnota 255. Pokud chceme používat všechny výstupní atributy a algoritmus nemá provádět filtraci atributů, nastavíme parametr na 0.

- **MAXIMUM_STATES**
  Nastavením tohoto parametru určujeme maximální počet stavů u atributů, které algoritmus podporuje. Při vytváření modelu je nastavena výchozí hodnota 100. Když má atribut více stavů, než je stanovená hodnota, využívají se nejpoužívanější stavy do tohoto omezujícího počtu. Se zbylými stavy pracuje jako s chybějícími.

- **SAMPLE_SIZE**
  Nastavením tohoto parametru určujeme počet objektů, které algoritmus používá při trénování. Při vytváření modelu je nastavena výchozí hodnota 10 000. Algoritmus používá buďto tento parametr, nebo HOLDOUT_PERCENTAGE, omezující je vždy ten parametr, který zpracovává méně dat. Parametr HOLDOUT_PERCENTAGE jsme si popsali na začátku této kapitoly.

3.5 Microsoft Sequence Clustering Algorithm

V češtině můžeme algoritmus Microsoft Sequence Clustering Algorithm (dále jen MSCA) označit jako shlukovací algoritmus sekvencí. Tento algoritmus vytváří skupiny označované jako shluky nebo klastry. Do jednotlivých shluků dává objekty, které mají podobné sekvence. Sekvencemi rozumíme sled nějakých po sobě jdoucích událostí. Příkladem může být procházení webových stránek. MSCA hledá nejčastější sekvence, z těch posléze vytváří klastry, ve kterých se nacházejí téměř identické sekvence. Tímto se MSCA liší od algoritmu MCA, kdy MCA vytváří klastry na základě podobných vlastností objektů.

Používá se pro analýzu sekvenčních dat, kdy chceme odhalit jednotlivé vlastnosti sekvencí. Tento algoritmus se doporučuje pro následující scénáře [2]:

- Procházení webových stránek.
- Pomocí logu událostí odhadnutí selhávání pevného disku, nebo selhání serveru.
- Transakční záznamy, ve kterém pořadí zákazník přidává zboží do nákupního košíku.
- Záznamy, které sledoval zákazník v průběhu času, odhadnutí zrušení služby nebo jiné nežádoucí výsledky.

3.5.1 Jak pracuje algoritmus

MSCA využívá Markovy řetězce (Markov chain [2]) k analýze sekvencí, kdy se snaží identifikovat jednotlivé sekvence, u kterých určuje pravděpodobnost. Podrobnější vysvětlení Markových řetězců nalezneme v dokumentu [11]. Využívá n-řádů Markových řetězců, označovány také jako n-order Markov chains.

Při vytváření shluků se využívá sekvenčních a nesekvenčních atributů. Počáteční rozdělení shluků se provede pomocí analýzy pravděpodobností přechodu, která měří vzájemné odlišnosti, a vypočítávají se vzdálenosti všech existujících sekvencí. Poté se pro vytváření shluků využívá EM metoda, kterou jsme si detailně popsali v kapitole 3.2.1. proto ji již nebudeme popisovat. Každý vytvořený klastr má svůj Markov model, který obsahuje kompletní sadu přechodových cest a pravděpodobností, tedy má svou přechodovou matici. Pro výpočet pravděpodobností sekvencí a atributů v konkrétním klastru se využívají Bayesova pravidla, více se můžeme dočíst o Bayesově větě v [8]. S vytvořenými klastry jsou spojeny jak sekvence, tak dodatečné atributy. Pomocí Markova modelu, který obsahuje každý klastr a pomocí Bayesových

pravidel můžeme vypočítat s jakou pravděpodobností je sekvence součástí daného klastru. Tyto výpočty již provádí algoritmus sám.

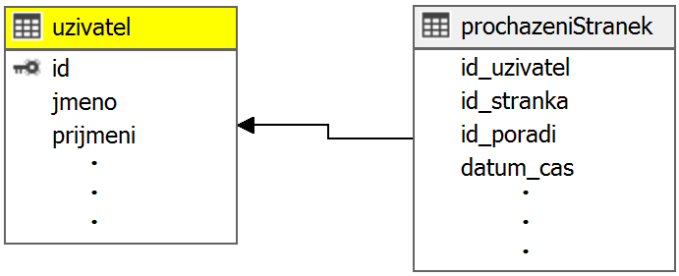
3.5.2 Požadavky

Pro úspěšné vytvoření modelu pro dolování dat pomocí algoritmu MSCA musí mít vstupní tabulky následující datovou strukturu:

- **Hlavní tabulka:**
  - Musí obsahovat klíčový sloupec.
  - Může obsahovat další atributy.

- **Vnořená tabulka:**
  - Musí obsahovat cizí klíč (e) z hlavní tabulky.
  - Musí obsahovat alespoň sekvenční sloupec, přičemž označit můžeme pouze jeden.
  - Může obsahovat další atributy.

Na obrázku 8 můžeme vidět ukázku struktury tabulky pro analýzu sekvenčních dat. Hlavní tabulka je s názvem „*uživatel*“, podřízená tabulka je „*prochazeniStranek*“. Podřízená tabulka obsahuje cizí klíč z hlavní tabulky, dále atribut *id_stranka* je jedinečný identifikátor stránky a atribut *id_poradi* je klíč sekvence.



Obrázek 8: Ukázkový diagram tabulek pro analýzu sekvenčních dat

MSCA musí obsahovat klíčový sloupec a vstupní sloupce, přičemž podporuje vstupní a předvídané sloupce následujícího typu:

- **Vstupní atributy:**
  Spojité, cyklické, diskrétní, diskretizované, klíčový sloupec, klíčový sloupec sekvence, tabulka, uspořádané.

- **Předvídatelné atributy:**
  Spojité, cyklické, diskrétní, diskretizované, tabulka, uspořádané.

3.5.3 Přizpůsobení

MSCA umožnuje upravovat výsledný model pomocí parametrů, které ovlivňují chování, výkon a přesnost výsledného modelu. Parametry jsou při vytváření modelu ve výchozím stavu, kdy každý parametr je nastaven ve výchozí hodnotě. Pomocí následujících parametrů můžeme měnit výsledný model:

- **CLUSTER_COUNT**
  Nastavením tohoto parametru určujeme přibližný počet klastrů, které mají být vytvořeny. Při vytváření modelu je nastavena výchozí hodnota 10. Jakmile nemůže být vytvořen model s přibližným počtem klastrů, MSCA vytvoří tolik klastrů, kolik jich je možných. Když nastavíme hodnotu na 0, algoritmus sám odhadne nejlepší počet klastrů.

- **MINIMUM_SUPPORT**
  Nastavením tohoto parametru určujeme minimální počet objektů, které jsou nutné k vytvoření klastru. Při vytváření modelu je nastavena výchozí hodnota 10.

- **MAXIMUM_SEQUENCE_STATES**
  Nastavením tohoto parametru určujeme maximální počet stavů, které mohou mít jednotlivé sekvence. Při vytváření modelu je nastavena výchozí hodnota 64. Jestliže nastavíme hodnotu větší než 100, výsledný model již nemusí obsahovat smysluplné informace.

- **MAXIMUM_STATES**
  Nastavením tohoto parametru určujeme maximální počet stavů u nesekvenčních atributů, které algoritmus podporuje. Při vytváření modelu je nastavena výchozí hodnota 100. Když má atribut více stavů, než je stanovená hodnota, využívají se nejpoužívanější stavy do tohoto omezujícího počtu. Zbylé stavy se označí jako chybějící.

4. Testovací data

V následující kapitole se seznámíme s testovacími daty, se kterými budeme pracovat. Testovací data budeme mít k dispozici v několika souborech a ve dvou formátech. Pomocí naprogramované aplikace provedeme jejich import do databázových tabulek. Pro následující kroky je důležité mít přístup k nějaké běžící instanci databáze minimálně ve verzi Microsoft SQL Server 2012 Standart Edition a vyšší. Verze Microsoft SQL Server 2012 Express Edition není žádoucí, neboť v této verzi nemůžeme provádět dolování dat, respektive neobsahuje službu Analysis Services. V databázi musíme mít dostatečná oprávnění, abychom mohli vytvářet, zapisovat, upravovat a případně mazat data z tabulek. Instalaci či konfiguraci databáze se zabývat nebudeme.

4.1 Popis testovacích dat

Celkem máme k dispozici několik datových souborů, v nichž máme uložena data, která nám poslouží pro výcvik či testování modelů pro dolování dat.

Deset souborů je v běžném textovém formátu s příponou „*txt*“. Jedná se o nesekvenční data. Tyto soubory obsahují různé síťové útoky, o kterých se můžeme dočíst v dokumentu [12]. My se nebudeme zabývat tím, jaké síťové útoky obsahují jednotlivé soubory. Co nás ale zajímá je to, že u každého záznamu je uvedeno, zda se jedná o síťový útok či ne. Názvy jednotlivých souboru jsou následující:

- class1.txt, class2.txt, class3.txt, class4.txt, class5.txt, testcl1.txt, testcl2.txt, testcl3.txt, testcl4.txt, testcl5.txt

Těchto 10 souborů rozdělíme do dvou skupin:

- **Učící skupina:**
  Zde bude prvních 5 souborů (class1.txt…class5.txt). Z těchto dat se modely pro dolování dat budou učit.

- **Testovací skupina:**
  Zde bude zbylých 5 souborů (testcl1.txt…testcl5.txt). U těchto dat budeme odhadovat, zda se jedná či nejedná o síťový útok.

V každém textovém souboru je na prvním řádku uvedeno kolik záznamů obsahuje, včetně počtu sloupců. Jak můžeme vidět na obrázku 9, tak soubor „class1.txt“ obsahuje 5 092 záznamů a 41 sloupců. Jednotlivé sloupce jsou odděleny tabulátorem. Datový soubor obsahuje ve skutečnosti 42 sloupců, v datovém souboru jsou totiž sloupce počítány od nuly. Poslední sloupec obsahuje záznam o tom, zda se jedná či nejedná o síťový útok. Tento sloupec nabývá pouze dvou stavů, a to nuly nebo jedničky. Nula znamená pro daný řádek, že se nejedná o síťový útok. Jednička znamená pro daný řádek, že se jedná o síťový útok. Obdobně je to u všech ostatních souborů, které byly uvedeny výše.

```
5092 41
0       0       0       1       0       0.0011  0       0       0       0       0
0       0       0       1       0       0.0001  0       0       0       0       0
0       0       0       1       0       0.0003  0       0       0       0       0
0       0       0       1       0       0.0003  0       0       0       0       0
0       0       0       1       0       0.0004  0       0       0       0       0
0       0       0       1       0       0.0004  0       0       0       0       0
0       0       0       1       0       0.0004  0       0       0       0       0
0       0       0       1       0       0.0008  0       0       0       0       0
0       0       0       1       0       0       0       0       0       0       0
```

Obrázek 9: Ukázka souboru class1.txt

V dokumentaci [12] je popsán význam jednotlivých sloupců, avšak tyto informace pro nás nejsou důležité, proto se jimi nebudeme zabývat. My si pouze řekneme, které sloupce jsou spojitého či diskrétního datového typu. Tyto datové typy jsou specifikovány v tabulce 5.

Tabulka 5: Proměnné datové typy

Proměnný datový typ	Sloupec
Diskrétní	1, 2, 3, 6, 11, 20, 21, 41
Spojitý	0, 4, 5, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40

Zbývá nám popsat již poslední soubor s názvem „*graph.gdf*“. V tomto souboru se nacházejí sekvenční data. Konkrétně se v tomto souboru nacházejí posloupnosti zobrazení stránek jednotlivých uživatelů. Na obrázku 10 můžeme vidět část datového souboru. Každý řádek, mimo prvního a posledního, obsahuje nějakou sekvenci procházení stránek. Jednotlivé sloupce jsou odděleny čárkou. Posloupnost průchodu webových stránek je oddělena středníkem.

Když se podíváme na první řádek, který začíná „*nodedef*“, tak zde máme definováno, jaké sloupce se v tomto datovém souboru nacházejí, včetně datových typů. Přehlednější zápis jednotlivých sloupců včetně vysvětlení se nachází v tabulce 6.

```
nodedef>name VARCHAR,label VARCHAR,color VARCHAR, width DOUBLE,
0,488;14;6326;398;6326;,'128,128,128',10,assignment view^811;as
1,488;,'128,128,128',10,assignment view^811;,488;
2,6326;6335;,'128,128,128',10,course view^světová ekonomika [op
3,6326;487;4190;4190;,'128,128,128',10,course view^světová ekon
4,6326;,'128,128,128',10,course view^světová ekonomika [opf-zs-
5,6326;6387;,'128,128,128',10,course view^světová ekonomika [op
6,6326;6368;488;,'128,128,128',10,course view^světová ekonomika
7,6326;454;2917;487;4190;,'128,128,128',10,course view^světová
```

Obrázek 10: Ukázka souboru graph.gdf

Tabulka 6: Popis jednotlivých sloupců souboru graph.gdf

Název	Datový typ	Popis
name	VARCHAR	Jedinečný identifikátor sekvence. V našem případě se jedná o jedinečný identifikátor nějakého uživatele.
label	VARCHAR	Jedinečné identifikátory jednotlivých stránek. Znak středník je oddělovačem sekvence navštívených stránek.
color	VARCHAR	Barva.
width	VARCHAR	Šířka.
meta0	VARCHAR	Obsahuje názvy jednotlivých stránek, které byly procházeny. Znak středník je oddělovačem sekvence navštívených stránek.
meta1	VARCHAR	Obsahuje stejná data jako sloupec label.

Nejdůležitější sloupce u tohoto datového souboru jsou:

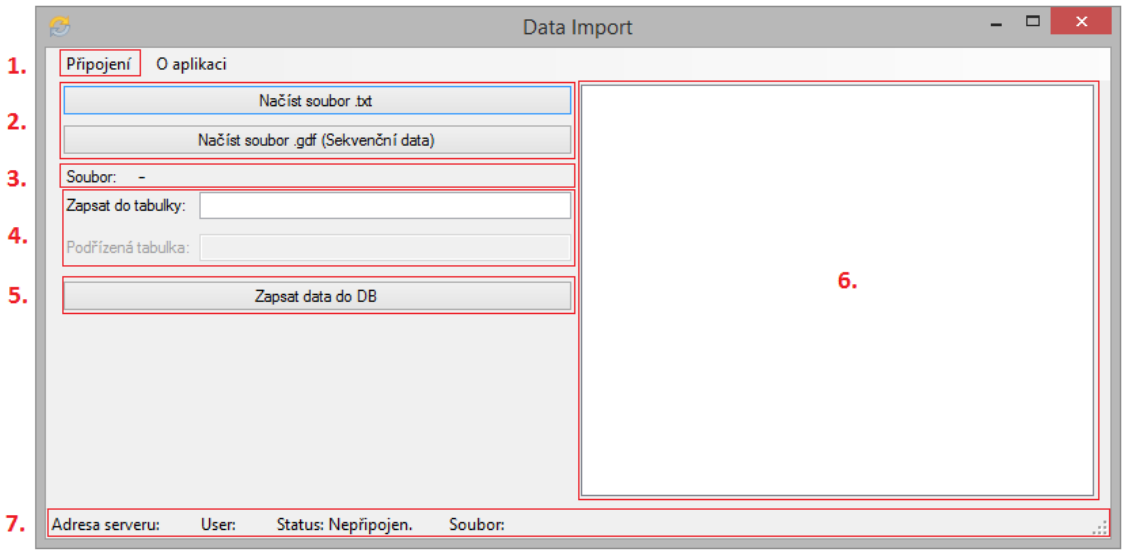
- name,
- label,
- meta0.

Když se podíváme na obrázek 10, který začíná hodnotou dvě, jedná se o sloupec s názvem *name*. Dále vidíme sloupec label, který obsahuje sekvenci o dvou zobrazených stránkách. Tyto stránky mají jedinečné identifikátory 6326 a 6335. To znamená, že uživatel začal na stránce s jedinečným identifikátorem 6326 a poté pokračoval na stránku 6335. Přeskočíme dva sloupce, které nesou informaci o barvě a šířce. Nyní máme sloupec s názvem *meta0*, který opět nese sekvenční informaci, jenom s tím rozdílem, že se zde nachází textový popis jednotlivých zobrazených stránek. Poslední sloupec meta1 nese opět stejné informace jako sloupec label.

4.2 Import testovacích dat

Nyní se budeme zabývat importem dat, která byla popsána v kapitole 4.1. Pro tento krok byla naprogramována aplikace s názvem „*Data import*“, která slouží k nahrání dat do tabulek databáze. V aplikaci se musíme přihlásit k běžící instanci databáze a následně budeme nahrávat data z jednotlivých souborů do databázových tabulek. Vytváření databázových tabulek za nás obstarává aplikace, čímž se nám velice usnadní nahrávání dat do tabulek databáze. Aplikace využívá hromadného nahrávání dat. Díky tomu se snižuje zatížení databáze a zrychluje se tím rychlost uložení dat do databáze.

Seznámíme se s tím, jak pracovat s aplikací pro import dat do databáze. Na obrázku 11 vidíme uživatelské rozhraní aplikace, včetně jejího popisu.



Obrázek 11: Popis uživatelského rozhraní aplikace pro import dat

Popis uživatelského rozhraní aplikace:

1. Kliknutím na toto tlačítko, nám vyskočí okno, ve kterém se budeme moci připojit k instanci databáze. Zadáváme:
    a. IP adresu,
    b. uživatelské jméno,
    c. heslo.

2. Zde vybíráme soubor, ze kterého chceme nahrát data do databázových tabulek. Zobrazí se nám okno, ve kterém zvolíme datový soubor, který chceme nahrát do databáze. Můžeme nahrávat soubory jak s příponou „*txt*“ tak i s „*gdf*“.

3. Zobrazení názvu zvoleného souboru.

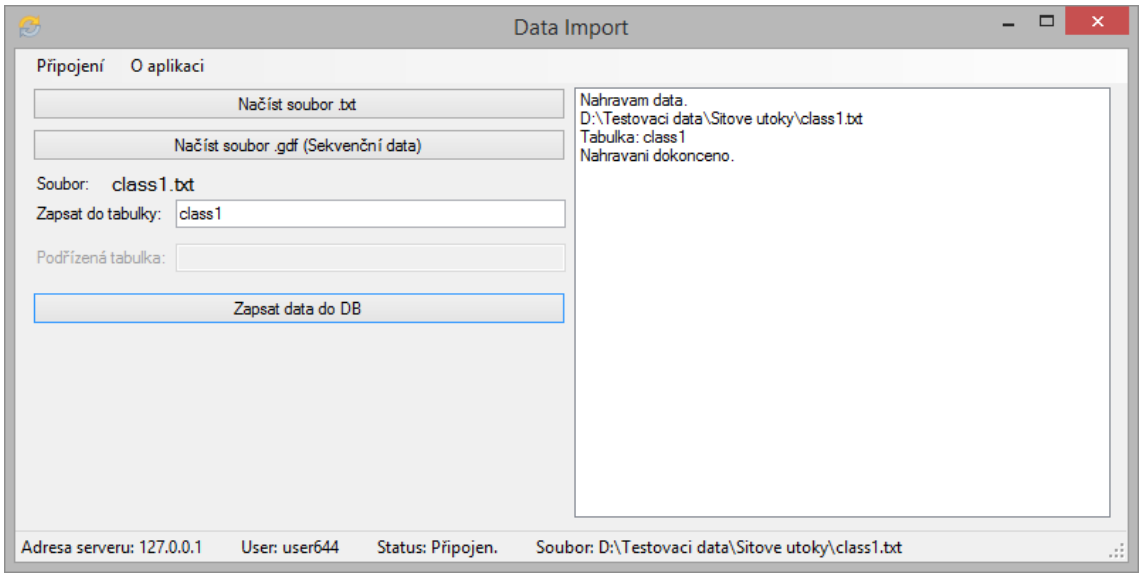
4. Zadání názvu tabulky, do které se mají uložit data ze zvoleného datového souboru. Automaticky se tento název tabulky vyplní podle názvu zvoleného datového souboru, tento název však můžeme změnit. Pokud zvolíme nahrávání dat ze souboru s příponou „*gdf*“, automaticky se také vyplní název podřízené tabulky, název je opět možno změnit.

5. Nyní můžeme provést import dat z datového souboru do tabulky v databázi.

6. Zobrazení informačních stavů o průběhu nahrávání dat do databáze.

7. Zde vidíme adresu databázového serveru, ke kterému se připojujeme, včetně uživatelského jména, stavu o tom zda jsme připojeni či ne. Jako poslední vidíme zvolený datový soubor, který jsme si vybrali.

Na obrázku 12, můžeme vidět úspěšné nahrání dat z datového souboru „*class1.txt*“ do tabulky v databázi s názvem „*class1*“. Takto nahrajeme postupně všechna data z datových souborů do databázových tabulek, přičemž doporučuji ponechat pro naše testování automaticky zadané názvy tabulek.



Obrázek 12: Úspěšné nahrání dat ze souboru *class1.txt*

Pro kontrolu se můžeme přihlásit k databázi pomocí Microsoft SQL Server Management Studio, ve kterém se připojíme ke stejné databázi se stejným uživatelským účtem a heslem jaké jsme použili v aplikaci pro import dat. Zkontrolujeme, zda máme vytvořené všechny tabulky včetně jejich obsahu. Můžeme využít následujících SQL dotazů, které vidíme v tabulce 7.

Tabulka 7: SQL dotazy pro kontrolu správnosti importovaných dat

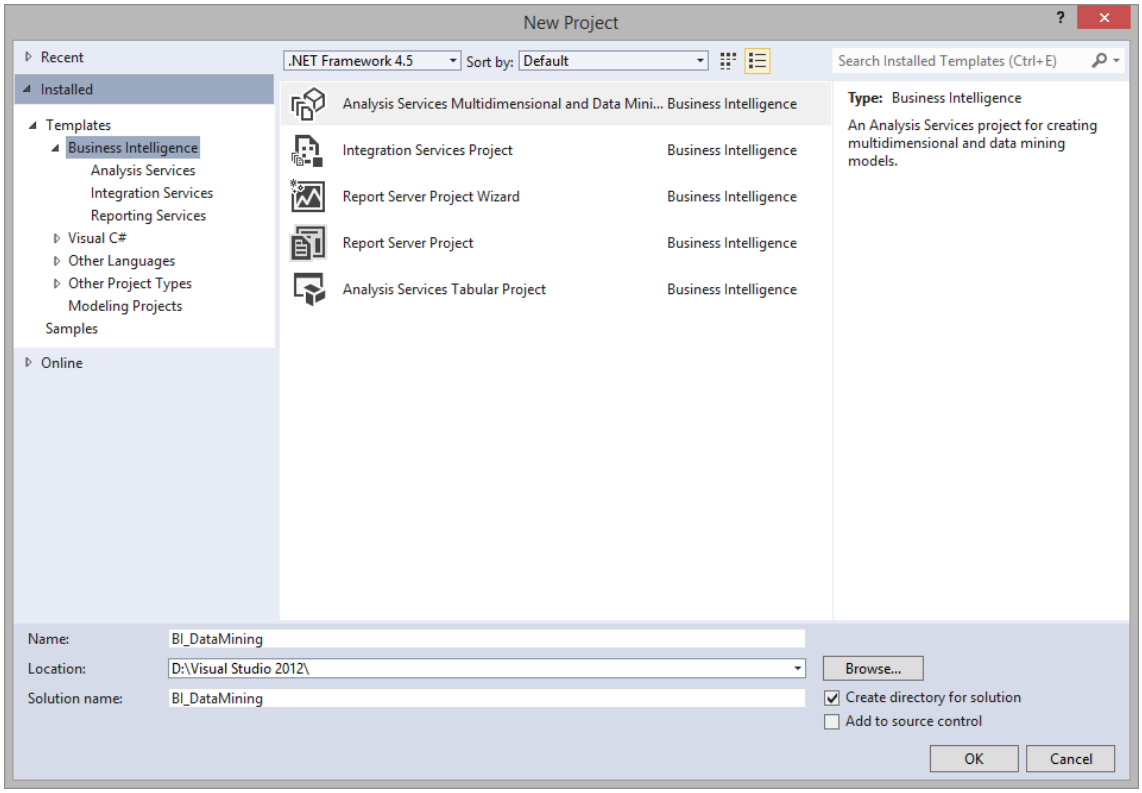
SQL dotaz	Popis
`SELECT * FROM nazev_tabulky;`	Výpis dat z tabulky, zkontrolujeme, zda nahraná data souhlasí s daty z datového souboru.
`SELECT COUNT(*) FROM nazev_tabulky;`	Zkontrolujeme, zda počet záznamů v tabulce souhlasí s počtem záznamů, který je uvedený v datovém souboru. Kontrolu provádíme vždy pro tabulku, ve které se nacházejí importovaná data z datového souboru.

5. Vytváření a úpravy modelů pro dolování dat

V následující kapitole se budeme věnovat vytváření a úpravě modelů pro dolování dat. Ukážeme si, kde a jak můžeme měnit jednotlivé parametry u vybraných algoritmů. Budeme využívat nástroje Microsoft Visual Studio 2012 s nainstalovaným doplňkem Microsoft SQL Server Data Tools – Business Intelligence for Visual Studio 2012. Tento doplněk se nachází na stránkách Microsoftu, který si musíme stáhnout a nainstalovat spolu s Microsoft Visual Studio 2012.

5.1 Založení projektu a počáteční nastavení

Spustíme Microsoft Visual Studio 2012 ve kterém si vytvoříme nový projekt. Zvolíme Business Intelligence, kde vybereme Analysis Services Multidimensional And Data Mining Project. Vyplníme název například „*BI_DataMining*“, popřípadě zvolíme umístění, kam se nám má uložit tento projekt, a potvrdíme. Vše můžeme vidět na obrázku 13.



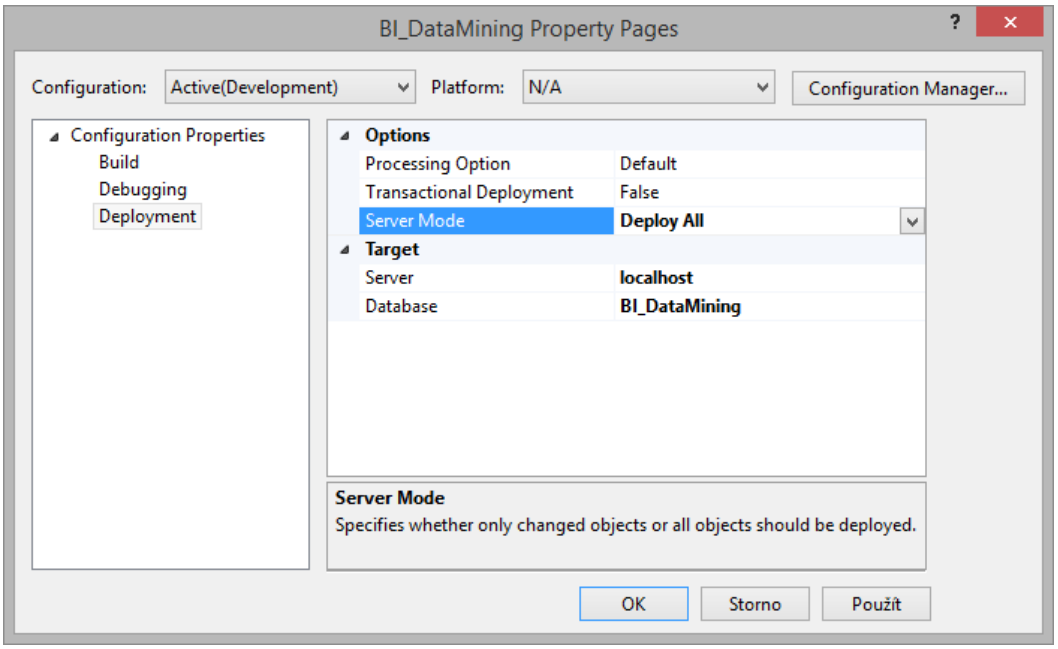
Obrázek 13: Vytváření projektu pro dolování dat

V základním nastavení projektu se modely pro dolování dat trénují až, když si jej chceme zobrazit. Toto trénování musíme ručně spustit, když si budeme chtít zobrazit nakonfigurovaný model.

Model je pouze nakonfigurován, ale není ještě sestaven, popřípadě je sestaven, ale upravili jsme konfiguraci modelu. Proto se nám zobrazí informační okno o tom, že musíme daný model nejprve sestavit či aktualizovat. Pokud máme více modelů pro dolování dat, toto trénování musíme ručně provádět zvlášť u každého modelu. Při prvním sestavení se model trénuje ze všech dat, které obsahuje trénovací množina. Při jeho dalším sestavení se model aktualizuje pouze již o upravené či přidané objekty v tabulce. Toto je lepší v tom případě, pokud projekt obsahuje mnoho modelů, které se trénují na velkém množství dat. Kdy každý model můžeme sestavovat či aktualizovat zvlášť. Takové to postupné sestavování modelů využijeme v případě, kdy nemáme dostatečně výkonný stroj, na kterém se budou modely trénovat.

Abychom nemuseli každý model pro dolování dat spouštět ručně, můžeme tento proces automatizovat. To provedeme tak, že při sestavování projektu se automaticky spustí trénování všech vytvořených modelů pro dolování dat. Nastavíme to pomocí parametru ve vlastnostech projektu, kdy klikneme na náš vytvořený projekt pravým tlačítkem myši a zvolíme „*Properties*“. Vybereme položku „*Deployment*“ a změníme parametr „*Server Mode*“ na „*Deploy All*“. Tímto se nám při sestavování modelů pro dolování dat budou trénovat všechny modely najednou. Při každém sestavení se modely trénují ze všech dat, které obsahuje trénovací množina. Toto nastavení můžeme vidět na obrázku 14.

Dále si všimněme parametru *Server* a *Database*. Jestliže se budeme připojovat na instanci databáze, která nám běží na stejném počítači a systému, je tento parametr nastaven na *localhost*. Pokud se budeme připojovat na jiný server, musíme zadat název serveru a instanci, kam se chceme připojit.



Obrázek 14: Nastavení parametru Server Mode

5.1.1 Nastavení připojení a definování zdrojových dat

Nyní musíme vytvořit připojení k databázi, ve které se nacházejí zdrojová data včetně služby Analysis Services. Nastavení připojení provedeme v našem vytvořeném projektu v Microsoft Visual Studio 2012 následujících krocích:

1. Ve vytvořeném projektu, klikneme pravým tlačítkem myši na položku „*Data Sources*“ a zvolíme „*New Data Source…*“.

2. Zobrazí se nám uvítací okno. Pokračujeme dále.

3. Vidíme okno pro definování připojení. Pokud jsme již někdy vytvářeli připojení k databázi, uvidíme jej zde. V našem případě se připojujeme poprvé, proto musíme pokračovat tlačítkem „*New…*“.

4. Nyní nastavujeme způsob připojení k instanci databáze. U položky „*Server name*“ můžeme zadat „*localhost“*, neboť nám běží databáze na stejném počítači. Popřípadě můžeme rozkliknout nabídku, kdy po chvilce konfigurátor nalezne název běžící instance databáze na našem počítači. Poté volíme ověřování pomocí autentizace SQL Server, kde zadáme svůj učet a heslo, například to, které jsme používali v kapitole 4.2. Jelikož se pod jednou instancí SQL Serveru může nacházet více běžících databází, můžeme definovat, kterou databázi chceme využívat. Jakmile máme vše vyplněno, můžeme provést kontrolu připojení pomocí tlačítka „*Test Connection*“. Pokračujeme potvrzovacím tlačítkem.

5. Opět vidíme okno pro definování připojení, již s námi nakonfigurovaným připojením, kdy jej označíme a pokračujeme dále.

6. Nyní definujeme pověření, jakým způsobem se bude služba Analysis Services připojovat k zdrojovým datům. Zvolíme „*Use the service account*“ a pokračujeme dále.

7. Posledním krokem je zadání názvu našeho nakonfigurovaného připojení k databázi. Ponecháme výchozí název „*Localhost*“ a dokončíme.

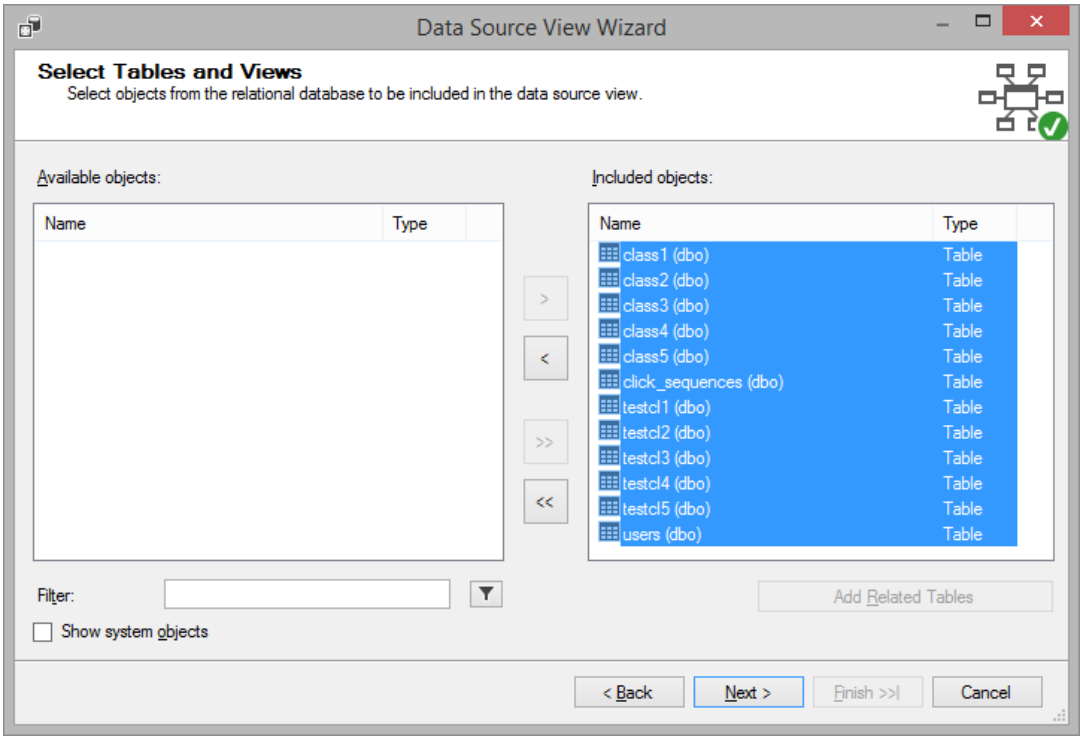
Jakmile jsme úspěšně dokončili nastavení připojení, definujeme nyní v Microsoft Visual Studio 2012 zdrojová data respektive zdrojové tabulky v těchto krocích:

1. Ve vytvořeném projektu, klikneme pravým tlačítkem myši na položku „*Data Source Views*“ a zvolíme „*New Data Source View…*“.

2. Zobrazí se nám uvítací okno. Pokračujeme dále.

3. Vybereme připojení, které jsme si již vytvořili. V našem případě se jedná o připojení s názvem „*Localhost*“ a pokračujeme.

4. Nyní definujeme, které tabulky a pohledy budeme používat. V levé části vidíme tabulky a pohledy které se nacházejí v databázi. V pravé části vidíme tabulky, se kterými budeme moci následně pracovat, vidět je můžeme na obrázku 15. Doporučením je vybírat pouze ty tabulky a pohledy, které potřebujeme. V našem případě si vybereme všechny tabulky, které jsme si vytvořili v kapitole 4.2 a pokračujeme dále.



Obrázek 15: Výběr tabulek a pohledů

5. Posledním krokem je zadání názvu tohoto nastavení výběru tabulek a pohledů. Ponecháme výchozí název „*Localhost*“ a dokončíme.

5.2 Vytváření modelů pro dolování dat

Postup vytváření modelů pro dolování dat si popíšeme v několika krocích. Jednotlivé kroky se u některých algoritmů budou lišit. Těmto změnám se budeme také věnovat. Vytváření modelů pro dolování dat se bude zabývat těmito algoritmy:

- Microsoft Association Rules Algorithm
- Microsoft Clustering Algorithm
- Microsoft Decision Trees Algorithm
- Microsoft Neural Network Algorithm

- Microsoft Sequence Clustering Algorithm

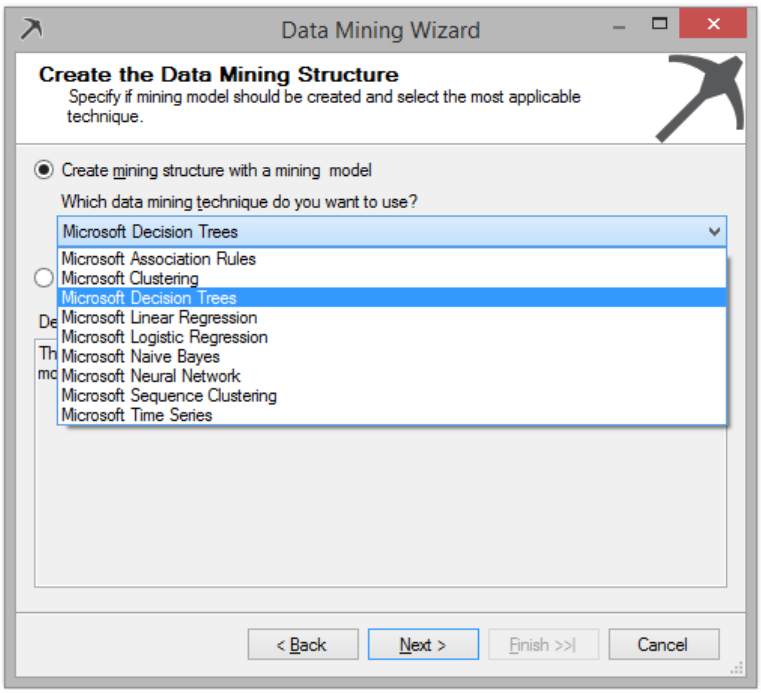
Ve vytvořeném projektu pod položkou „*Mining Structures*“ se budou nacházet vytvořené struktury pro dolování. Z vytvořené struktury následně model pro dolování dat využívá informace, jako je definování vstupních dat respektive tabulek a atributů. V těchto strukturách je definováno:

- Hlavní tabulka, popřípadě vnořená tabulka.
- Klíčový, vstupní, odhadovaný či ignorovaný sloupec.
- Nastavení datových typů jednotlivých atributů.

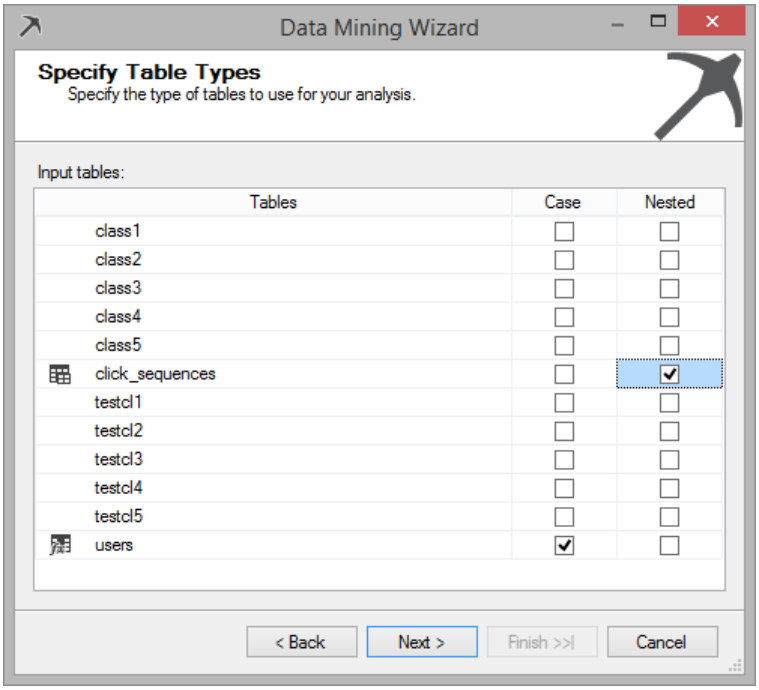
V těchto strukturách budou následně jednotlivé modely pro dolování dat. Postup vytváření modelu pro dolování dat v Microsoft Visual Studio 2012 je následující:

1. Ve vytvořeném projektu, klikneme pravým tlačítkem myši na položku „*Mining Structures*“ a zvolíme „*New Mining Structure…*“.
2. Zobrazí se nám uvítací okno vytváření modelu pro dolování dat. Pokračujeme dále.
3. Nyní můžeme definovat, zda budeme vytvářet model z relační databáze či OLAP kostky. Necháme zvolenou možnost vytváření modelu z relační databáze a pokračujeme dále.
4. Poté volíme, zda chceme vytvořit strukturu již s modelem či bez něj. Ponecháme vytvoření struktury pomocí modelu pro dolování dat, u kterého si zvolíme, jaký algoritmus chceme použit. Přehled algoritmů můžeme vidět na obrázku 16. Kdy budeme postupně volit algoritmy, které jsme si uvedli výše.
5. Dále vybíráme zdrojová data, ve kterých se nacházejí zdrojové tabulky a pohledy. Volíme naše nakonfigurované zdrojové tabulky a pohledy, které jsme si uložili pod názvem *Localhost*, v kapitole 5.1.1.
6. Vybíráme zdrojové tabulky, přičemž můžeme zvolit pouze jednu hlavní tabulku, tu si označíme ve sloupci *Case*. Pokud potřebujeme přidat k této tabulce podřízenou tabulku, označíme ji ve sloupci *Nested*. Pro vybrané algoritmy budeme používat postupně tabulky s názvy *class1, class2, class3, class4* a *class5*.

   Pokud budeme vytvářet model pro Microsoft Sequence Clustering, zvolíme si vstupní tabulky tak, jako je můžeme vidět na obrázku 17. Kdy hlavní tabulka je *users* a podřízenou tabulkou ve které se nacházejí sekvenční data je *click_sequences*.



Obrázek 16: Přehled algoritmů pro dolování dat

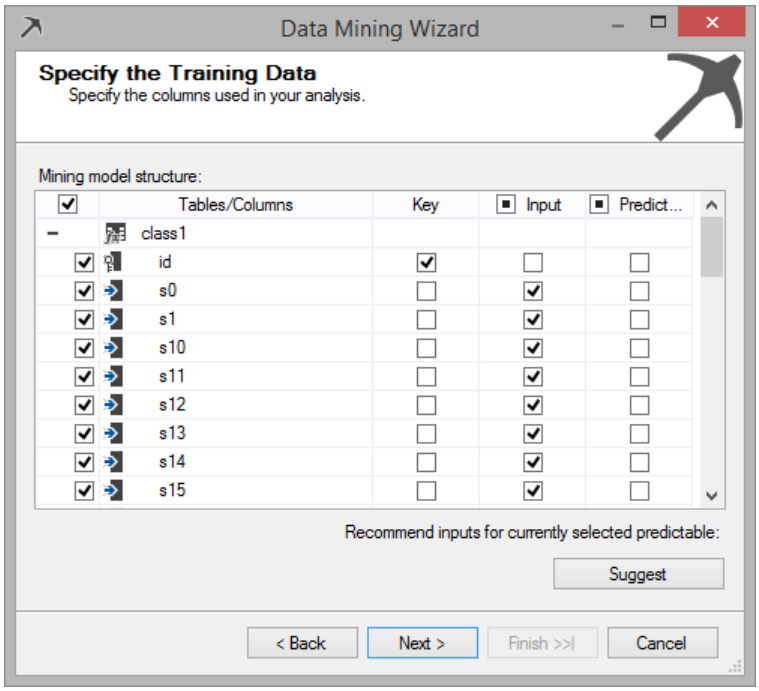


Obrázek 17: Označení tabulek pro algoritmus Microsoft Sequence Clustering

7. Nyní specifikujeme jednotlivé sloupce použité tabulky. Pro modely mimo Microsoft Sequence Clustering, definujeme klíčový, vstupní a předvídaný sloupec či sloupce.

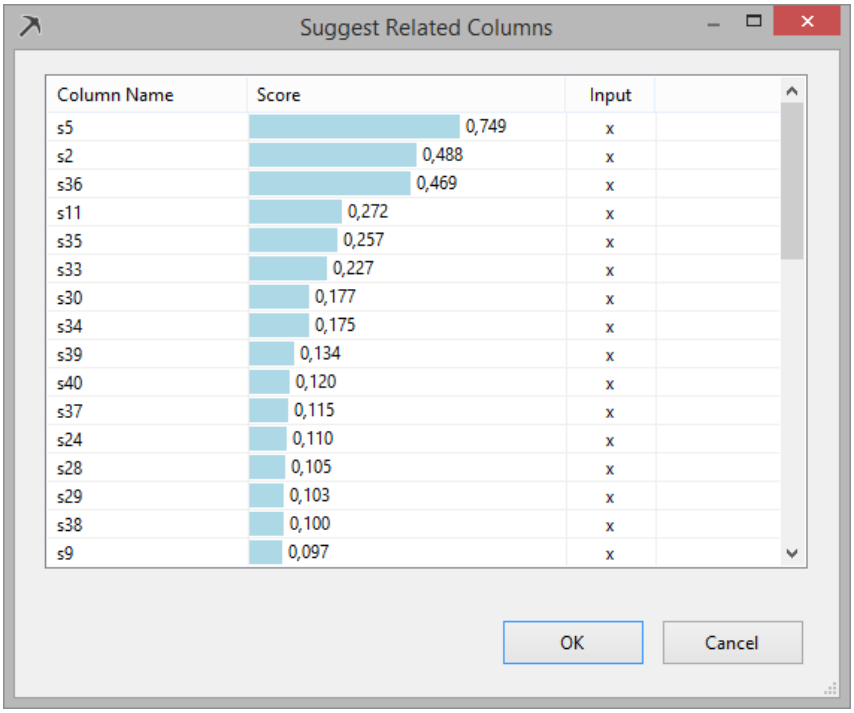
Klíčový sloupec se označí automaticky. Jako vstupní sloupce označíme s0-s40. Jako odhadovaný sloupec označíme s41. Část tohoto nastavení můžeme vidět na obrázku 18.

Všimněme si tlačítka *Suggest*. Pokud máme zvolený odhadovaný sloupec, tak stisknutím tohoto tlačítka se provede analýza dat a identifikují se sloupce, které nejvíce korelují s předvídaným sloupcem. Ve výsledku se nám zobrazí doporučení pro výběr vstupních sloupců, jak můžeme vidět na obrázku 19. Když potvrdíme tento dialog pro doporučení výběru vstupních sloupců, automaticky se tyto sloupce označí. My tuto analýzu nyní nebudeme používat.

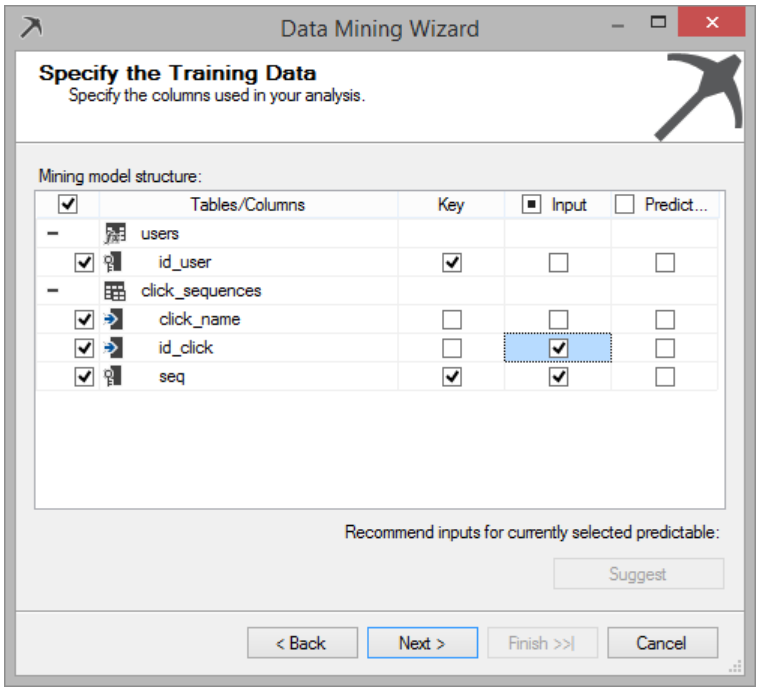


Obrázek 18: Označení sloupců

Při vytváření modelu pomocí Microsoft Sequence Clustering se nám automaticky označí klíčový sloupec z tabulky *users*. Dále musíme definovat vstupní sloupce a klíčový sloupec, které jsou v tabulce *click_sequences*. Tato tabulka obsahuje tři sloupce, které můžeme označit. Dva sloupce jsou sekvenční, přičemž označit můžeme pouze jeden, zvolíme *id*_click. Poslední sloupec *seq* je klíčem sekvence a proto jej označíme jako klíčový. Nyní nebudeme označovat žádný sloupec, který bychom chtěli odhadovat. Tento celý postup můžeme vidět na obrázku 20.



Obrázek 19: Výsledek doporučení výběru vstupních atributů



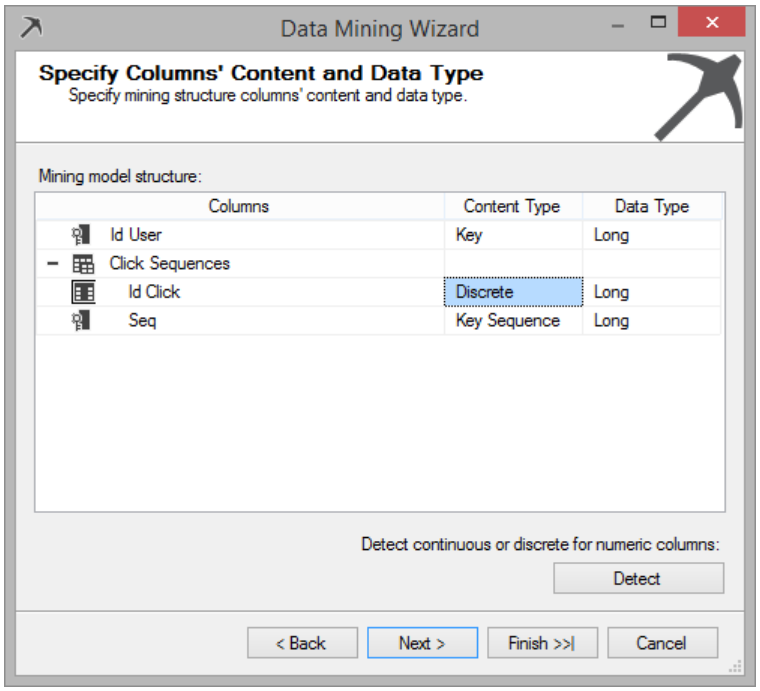
Obrázek 20: Označení sloupců u modelu Microsoft Sequence Clustering

8. V tomto kroku provedeme definování, zda se jedná o spojitý či diskrétní sloupec, včetně nastavení datového typu. Ve výchozím nastavení jsou všechny sloupce nastaveny na spojitý datový typ. Máme zde možnost automatické detekce, zda se jedná o spojitá či diskrétní data, kterou můžeme vyvolat tlačítkem *Detect*. Tuto volbu nyní ale využívat nebudeme, neboť se budeme řídit tabulkou 5, která se nachází v kapitole 4.1, kdy jednotlivé sloupce nastavíme sami. Sloupec s číslem *0* odpovídá v tabulce sloupci s označením *s0*. Dále pak můžeme nastavit, o jaký datový typ se jedná, na výběr máme z těchto možností:
    - Binary,
    - Boolean,
    - Date,
    - Double,
    - Long,
    - Text.

   V našem případě můžeme všude ponechat výchozí nastavení datových typů.

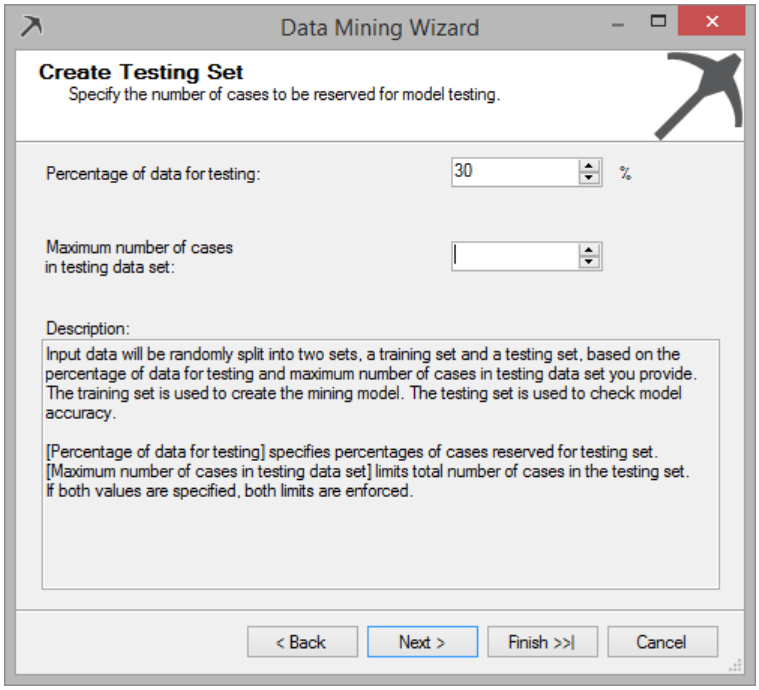
   U algoritmu Microsoft Association Rules jsou ve výchozím nastavení všechny sloupce nastaveny na diskretizovaný (*Discretized*) datový typ. Nastavíme pouze sloupce podle tabulky 5, která se nachází v kapitole 4.1, na diskrétní datové typy.

   U algoritmu Microsoft Sequence Clustering musíme nastavit pouze sloupec *id_click* na diskrétní datový typ. Tento postup můžeme vidět na obrázku 21. Všimněme si, že sloupec *seq* je nastaven jako klíč sekvence.



Obrázek 21: Nastavení sloupců u algoritmu Microsoft Sequence Clustering

9. Nyní zadáváme, kolik dat se použije pro testování modelu, zbylá data budou použita k trénování modelu. Jedná se tedy o poměr rozdělení dat mezi testovací a učící množinou, které můžeme vidět na obrázku 22. Toto rozdělení zadáváme v procentech anebo maximálním počtem objektů v data setu v testovací množině. Pokud vyplníme obě položky, budou se používat oba limity. Toto rozdělení bude omezovat vždy ten limit, který vybírá méně dat. Například zvolíme-li limit 30% a maximální počet objektů 1 000 a v tabulce se bude nacházet 1 milion záznamů, tak do testovací množiny bude patřit pouze 1 000 objektů. Jakmile není vyplněna položka maximálního počtu objektů v data setu, znamená to, že zde není žádný limit. Výběr konkrétních objektů do jednotlivých množin se provádí náhodně. Vše ponecháme ve výchozím nastavení.



Obrázek 22: Nastavení velikosti testovací množiny

10. Zobrazí se nám poslední dialog, ve kterém můžeme vyplnit *Mining structure name*. Pod tímto názvem se nám objeví vytvořená struktura ve vytvořeném projektu pod položkou *Mining Structures*. Například pro strukturu, kde se používá vstupní tabulka class1, můžeme vyplnit název Class1. Dále pak máme *Mining model name*, ve kterém zadáme název modelu pro dolování dat, tento název se objeví ve vytvořené struktuře. Například když vytvoříme model pro dolování dat s algoritmem Microsoft Decision Trees pro vstupní tabulku *class1*, zadáme název „*Class1 DecisionTrees*“. Obdobně budeme zadávat názvy dalších vytvořených modelů podle vstupní tabulky a použitého algoritmu. Tímto se dostáváme ke konci vytváření modelu pro dolování v datech.

Jedna struktura může obsahovat více modelů pro dolování dat. Například v našem případě můžeme do jedné struktury vložit tyto algoritmy pro dolování dat:

- Microsoft Clustering Algorithm
- Microsoft Decision Trees Algorithm
- Microsoft Neural Network Algorithm

Postup přidání modelu do vytvořené struktury je popsána v kapitole 5.3.1.

Pro ověření, že jsou modely pro dolování dat nakonfigurovány správně, otevřeme si nabídku *Build*, kterou máme v horní části *Microsoft Visual Studia* a provedeme nasazení modelů pomocí *Deploy Solution*. Zobrazí se nám průběh sestavování modelů, ve kterém můžeme vidět čas spuštění a dokončení modelu, tedy jak dlouho se jednotlivé modely vypočítávaly. Dále jsme informování o úspěchu či neúspěchu sestavování modelu. Pokud dojde k neúspěchu, můžeme si jednotlivé chyby zobrazit v záložce *Error List*. Často se nám zobrazí chyba u modelů, kde se používá Microsoft Assiociation Rules. Tyto chyby se budou zobrazovat pro ty sloupce, které obsahují příliš málo různých hodnot. O jaký sloupec a model se jedná, zjistíme ve výpisu chyb. Proto je zapotřebí dané sloupce nastavit na hodnotu diskrétního datového typu. Sloupce s touto hodnotou poznáme z výpisu chyb. Jak lze provést tyto změny si řekneme v kapitole 5.3.1.

5.3 Úprava modelů pro dolování dat

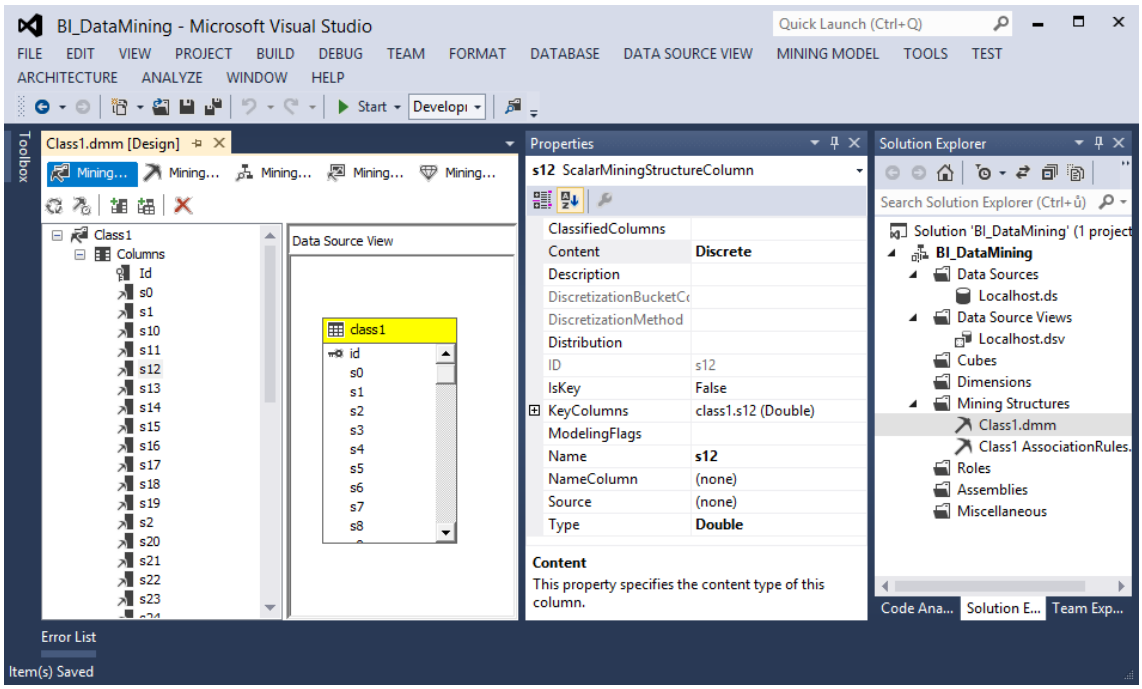
U vytvořeného modelu pro dolování dat můžeme provádět úpravu datové struktury či nastavování parametrů u použitého algoritmu pro dolování dat. Úprava datové struktury se nám může hodit například, když jsme nesprávně nakonfigurovali model pro dolování dat, ve kterém potřebujeme pouze poupravit nějaký atribut. Abychom nemuseli celý model pro dolování dat vytvářet znovu, stačí pouze provést úpravu již vytvořené datové struktury.

Další důležitou úpravou modelu pro dolování dat, je nastavení parametrů u použitého algoritmu pro dolování dat.

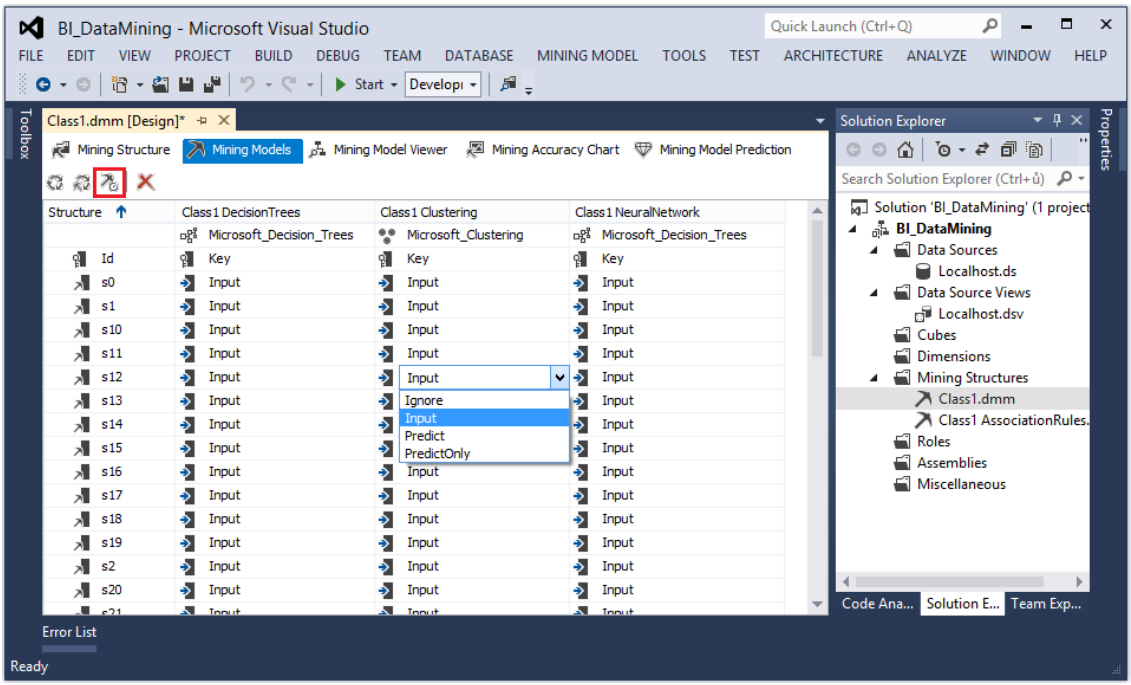
5.3.1 Úprava datové struktury

Při zobrazení nějaké vytvořené struktury, pod záložkou *Mining Structure* uvidíme název tabulky, ze které se model trénuje, včetně jejich jednotlivých sloupců. V levé části vidíme jednotlivé sloupce. Při kliknutí na jakýkoliv sloupec, můžeme v záložce *Properties* měnit nastavení zvoleného sloupce. Přehled nastavení můžeme vidět na obrázku 23.

V záložce *Mining Models* můžeme přidávat do již vytvořené struktury modely pro dolování dat. Provádí se to tak, že klikneme v horní části na prvek *Create a related mining model*, který je na obrázku 24 v červeném rámečku. Tímto se nám otevře dialog, ve kterém zadáme název modelu a vybereme algoritmus, který chceme použít v takto přidávaném modelu. U každého algoritmu, můžeme měnit nezávisle na celé struktuře, zda se jedná o vstupní či odhadovaný atribut. Pokud nastavíme atribut na hodnotu *Ignore*, nebude se k trénování modelu používat vůbec.



Obrázek 23: Přehled dodatečného nastavení sloupce s12 ve struktuře Class1



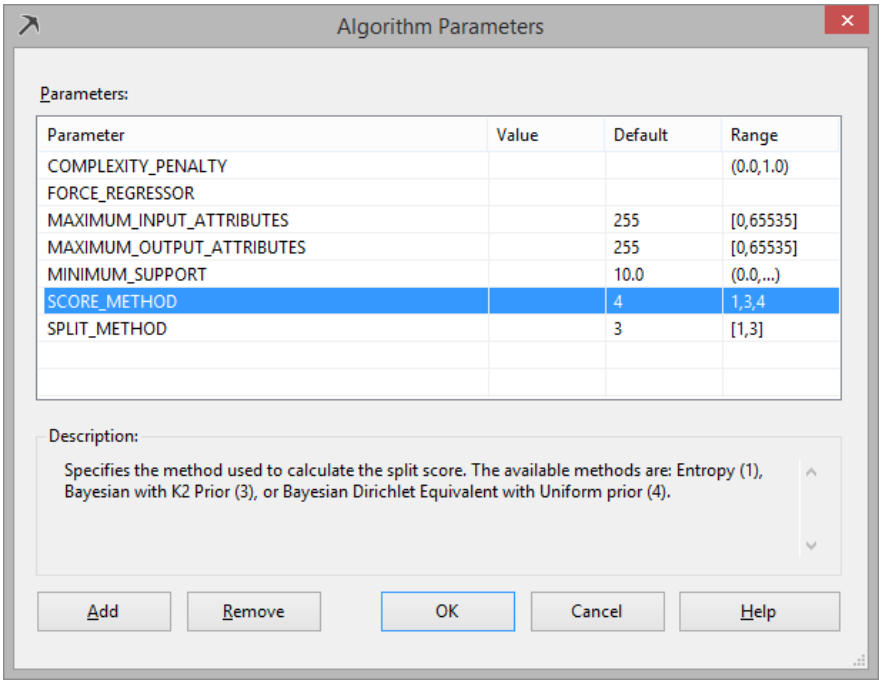
Obrázek 24: Přehled vytvořených modelů ve struktuře Class1

5.3.2 Úprava parametrů

Při zobrazení nějaké vytvořené struktury, pod záložkou *Mining Structure* uvidíme název tabulky, ze které se model trénuje, včetně jejich jednotlivých sloupců. Při označení tabulky můžeme v záložce *Properties* nastavovat několik parametrů. Nejdůležitějšími parametry jsou pro nás ty, které byly popsány na začátku kapitoly 3. Jedná se o tyto parametry:

- **HoldoutMaxCases**
- **HoldoutMaxPercent**
- **HoldoutSeed**

Dále můžeme měnit parametry jednotlivých algoritmů pro dolování dat. O jaké parametry se jedná u jednotlivých algoritmů, jsme se dočetli v kapitole 3. Pokud chceme změnit parametry nějakého algoritmu modelu, tak si musíme zobrazit strukturu, která obsahuje nějaký model pro dolování dat. Zvolíme si záložku *Mining Models*, tak jak to můžeme například vidět na obrázku 24. Když chceme změnit parametry algoritmu Microsoft Decision Trees pro model *class1_DecisionTrees*, klikneme na jakýkoliv prvek ve sloupci pro tento model a pravým tlačítkem myši zvolíme možnost *Set Algorithm Parameters…* Zobrazí se nám okno, ve kterém můžeme měnit jednotlivé parametry ve sloupci s názvem *Value*. Uvidíme zde, jaké jsou výchozí hodnoty (sloupec *Default*) jednotlivých parametrů a v jakém intervalu (sloupec *Range*) se mohou nastavovat. V dolní části je popis označeného parametru. Vše můžeme vidět na obrázku 25. Zobrazené parametry se budou lišit podle algoritmu, který chceme upravit.



Obrázek 25: Nastavení parametrů u algoritmu Microsoft Decision Trees

6. Analýza vytvořených modelů pro dolování dat a vytváření odhadů

V této kapitole si popíšeme, jak si můžeme zobrazit sestavený model pro dolování dat. Řekneme si, jaké informace můžeme vyčíst při prohlížení modelů a jaké jsou možnosti tohoto prohlížení. Prohlížení modelů a informace které se nám zobrazují, se liší na základě zvoleného algoritmu, pomocí kterého se sestavoval výsledný model pro dolování dat.

Dále provedeme analýzu úspěšnosti předpovídání odhadů jednotlivých modelů, které odhadují, zda se jedná o sítový útok či nikoliv. Pro každý algoritmus vytvoříme dva modely, kdy v prvním případě necháme všechny parametry v základním nastavení a provedeme analýzu úspěšnosti odhadů. V druhém případě upravíme některé parametry, abychom dosáhli lepších výsledků odhadů, tedy aby se snížila chybovost těchto odhadů.

6.1 Prohlížení sestaveného modelu

Prohlížet si sestavený model můžeme při otevření jakékoliv struktury, kde poté zvolíme záložku *Mining Model Viewer*. Prohlížení sestaveného modelu se liší na základě algoritmu, který se používá pro dolování dat, přičemž se může skládat z dalších různých záložek. Všechny prohlížené modely mají tyto společné možnosti:

- **Refresh viewer content:**
  Slouží opětovnému načtení při prohlížení modelu pro dolování dat.

- **Mining Model:**
  Pokud struktura obsahuje více modelů pro dolování dat, zde si zvolíme, který model si chceme zobrazit.

- **Viewer:**
  Zde si nastavíme volbu prohlížení vybraného modelu pro dolování. Máme na výběr ze dvou možností, kdy první možností je zobrazit si model v grafické podobě. Název tohoto zobrazení záleží na typu algoritmu, ze kterého je sestaven výsledný model. Například pro Microsoft Decision Trees Algorithm je název tohoto zobrazení *Microsoft Tree Viewer*. Druhou možností je zobrazení modelu pomocí obecného stromového zobrazení podle standardizovaného schématu nazývaný jako *Microsoft Generic Content Tree Viewer*.

  Microsoft Generic Content Tree Viewer zobrazuje detailní informace modelu pro dolování dat v standardizovaném HTML režimu tabulek. Toto zobrazení je užitečné, neboť odhaluje základní strukturu modelu, jako jsou podrobností o koeficientech, rozdělení hodnot atd. Obsah zobrazující se v tabulkách, se liší podle zvoleného algoritmu, sloupců, pravidel, vlastností, uzlů a vzorců. Jak správně interpretovat informace pro každý typ algoritmu se můžeme detailněji dočíst v dokumentaci [2].

  Prohlížení sestavených modelů budeme provádět pomocí grafické podoby.

6.1.1 Microsoft Association Rules Algorithm

Zobrazený model, který je sestaven pomocí Microsoft Association Rules Algorithm obsahuje tyto záložky *Rules, Itemsets* a *Dependency Network.*

Záložka Rules
Zde se zobrazují použitá pravidla, která algoritmus nalezl v datech. Tato pravidla se pak používají k vytvoření předpovědi či odhadnutí výsledku. Při prohlížení modelu můžeme nastavovat parametry, pomocí kterých filtrujeme zobrazená pravidla. Popis rozhraní záložky *Rules* je následující, ve které se nacházejí tyto prvky:

- **Minimum probability:**
  Nastavení minimální hodnoty pravděpodobnosti pravidel pro zobrazení při prohlížení. Zvýšením této hodnoty snížíme počet zobrazených pravidel.

- **Minimum importance:**
  Nastavení minimální hodnoty významu pravidel pro zobrazení při prohlížení. Zvýšením této hodnoty snížíme počet zobrazených pravidel.

- **Filter Rule:**
  Filtrování pravidel při prohlížení podle zadaného kritéria. Můžeme zadávat Microsoft .NET regulární výrazy. Jak správně zadávat regulární výrazy se dočteme v dokumentaci [13].

- **Show:**
  Nastavení výpisu pravidel, respektive, které informace o pravidlech budou vypsány. Na výběr máme z těchto možností:
  - Zobrazení názvu atributu a hodnoty (Show the attribute name and value)
  - Zobrazení pouze hodnoty atributu (Show the attribute value only)
  - Zobrazeni pouze názvu atributu (Show the attribute name only)

- **Show long name:**
  Zobrazení celého názvu pravidla.

- **Maximum rows:**
  Omezení maximálního počtu zobrazených pravidel.

- **Probability:**
  Tento sloupec v grafu zobrazuje pravděpodobnosti pro každé pravidlo. Kliknutím na záhlaví sloupce se provede seřazení podle pravděpodobnosti.

- **Importance:**
  Tento sloupec v grafu zobrazuje význam pro každé pravidlo. Kliknutím na záhlaví sloupce se provede seřazení podle významu.

- **Rule:**
  Tento sloupec v grafu zobrazuje textový popis pro každé pravidlo, podle zvoleného formátu nastaveného pomocí parametrů *Show* a *Show long name*.

Záložka Itemsets
Zde se zobrazují časté pravidla, které tento model pro dolování dat obsahuje. Při prohlížení itemsets můžeme nastavovat parametry, pomocí kterých filtrujeme zobrazené itemsets. Popis rozhraní záložky *Itemsets* je následující, ve které se nacházejí tyto prvky:

- **Minimum support:**
  Nastavení minimální hodnoty podpory, která itemset musí obsahovat při prohlížení. Zvýšením této hodnoty snížíme počet zobrazených itemsets.

- **Minimum itemset size:**
  Nastavení minimálního počtu položek, které itemset musí obsahovat při prohlížení. Zvýšením této hodnoty snížíme počet zobrazených itemsets.

- **Filter Itemset:**
  Filtrování itemsets při prohlížení podle zadaného kritéria. Můžeme zadávat jako Microsoft .NET regulární výrazy, jak správně zadávat regulární výrazy se dočteme v dokumentaci [13].

- **Show:**
  Nastavení jak se nám budou vypisovat itemset, respektive které informace o itemset budou vypsány. Na výběr máme z těchto možností:
  - Zobrazení názvu atributu a hodnoty (Show the attribute name and value)
  - Zobrazení pouze hodnoty atributu (Show the attribute value only)
  - Zobrazeni pouze názvu atributu (Show the attribute name only)

- **Show long name:**
  Zobrazení celého názvu itemset.

- **Maximum rows:**
  Omezení maximálního počtu zobrazených itemset. Ve výchozím nastavení jsou zobrazené itemsets seřazeny podle podpory v sestupném pořadí.

- **Support:**
  Tento sloupec zobrazuje podporu pro každý itemset.

- **Size:**
  Tento sloupec zobrazuje počet položek, které jsou v itemset.

- **Itemset:**
  Tento sloupec zobrazuje popis jednotlivých itemset, podle zvoleného formátu nastavených pomocí parametru *Show* a *Show long name*.

Záložka Dependency Network
Zobrazuje grafický pohled na všechny atributy, které model pro dolování dat obsahuje. Ukazuje, jak jsou jednotlivé atributy propojené. Tato záložka obsahuje intuitivní ovládací prvky, které popisovat nebudeme. Hrany v grafu představují propojení mezi dvěma uzly popřípadě atributy. Posuvníkem, který se nachází v ovládacích prvcích, můžeme postupně všechny hrany odstranit, anebo si je ponechat všechny zobrazené. Hrany představují důležitost pravidel, která jsou spojena dvěma itemsets. Ty hrany, které se odstraňují jako první, reprezentují menší důležitost hran mezi odhadovaným a vstupním atributem.

6.1.2 Microsoft Clustering Algorithm

Zobrazený model, který je sestaven pomocí Microsoft Clustering Algorithm obsahuje tyto záložky *Cluster Diagram*, *Cluster Profiles*, *Cluster Characteristics* a *Cluster Discrimination*.

Záložka Cluster Diagram
Poskytuje grafický přehled všech klastrů, které tento model obsahuje. Tato záložka poskytuje intuitivní ovládací prvky, které popisovat nebudeme. Jednotlivé klastry smíme přejmenovávat. Při najetí myší na klastr se nám zobrazí počet prvků nacházejících se v klastru. Popíšeme si pouze tyto nejdůležitější prvky:

- **Density:**
  Odstínové zobrazení počtu prvků v klastru. O jaký prvek či prvky se jedná, závisí na nastaveném parametru *Shading Variable*.

- **Links:**
  Slouží k nastavení zobrazení nejsilnější vztahů mezi jednotlivými klastry. Snížením se ponechávají pouze nejsilnější vazby mezi klastry.

- **Shading Variable:**
  Výběr atributu, který je zastoupen v jednotlivých klastrech popřípadě zvolení celé populace (*Population*).

- **State:**
  Zvolení jednoho stavu, který se použije pro stínování v grafu.

Záložka Cluster Profiles
Poskytuje celkový přehled klastrů, které tento model obsahuje. Zobrazují se zde všechny atributy, společně s distribucí atributů v každém klastru. Popis rozhraní záložky *Cluster Profiles* je následující, ve které se nacházejí tyto prvky:

- **Show Legend:**
  Zobrazení obsažených barev ve sloupci *States* pro zvolený klastr.

- **Histogram Bars:**
  Nastavení zobrazení počtu stavů v histogramu. Pokud existuje více stavů než je nastavený počet, tak se zobrazí pouze stavy, které mají nejvyšší pravděpodobnost.

- **Attributes:**
  Seznam sloupců, které se nacházejí v jednotlivých klastrech.

- **States:**
  Poskytuje informace o barvách každého stavu v řádku klastru. Rozdělení spojitých číselných hodnot nám označuje posuvník s diamantem.

- **Cluster Profiles:**
  Tato část obsahuje sloupec pro každý klastr, který se nachází v tomto modelu. Pro každý atribut histogram ukazuje rozložení jednotlivých hodnot atributů pro daný klastr.

Záložka Cluster Characteristics:
Umožnuje prozkoumávání jednotlivých klastrů, které model obsahuje. Popis rozhraní záložky *Cluster Characteristics* je následující, ve které se nacházejí tyto prvky:

- **Cluster:**
  Zvolení klastru, který chceme zobrazit. Další možností je zvolení *Population (All)*, kde uvidíme rozložení atributů v celém modelu jako celku.

- **Characteristics for <cluster>:**
  Zobrazuje následující sloupce, ve kterých jsou obsaženy informace zvoleného klastru:
  - **Variables** – obsahuje atributy modelu, které se nacházejí ve zvoleném klastru.
  - **Values** – obsahuje aktuální hodnoty atributů, které se nacházejí ve zvoleném klastru.
  - **Probability** – zobrazení síly zastoupení atributu a jeho hodnoty ve zvoleném klastru. Při najetí myši se zobrazí procentuální výpis. Ukazuje, s jakou pravděpodobností bude atribut-hodnota patřit do tohoto klastru.

Záložka Cluster Discrimination
Poskytuje možnost porovnat dva klastry, které obsahuje model. Můžeme zde vidět atribut a hodnotu jak jsou zastoupeny v porovnávaných klastrech. Popis rozhraní záložky *Cluster Discrimination* je následující, ve které se nacházejí tyto prvky:

- **Cluster 1:**
  Zvolení klastru, který chceme porovnat s jiným klastrem.

- **Cluster 2:**
  Zvolení druhého klastru, se kterým se provede porovnání k prvnímu zvolenému klastru. Porovnat také smíme doplněk k prvnímu klastru, atribut-hodnota nebude součástí zvoleného klastru (Cluster 1).

- **Discrimination scores for <cluster 1> and <cluster 2>:**
  Zobrazuje následující sloupce, ve kterých jsou obsaženy informace o atributu a hodnotě tykajících se dvou zvolených klastrů:
  - **Variables** – atribut.
  - **Values** – hodnota atributu.
  - **Favors <Cluster 1>** – sloupcový graf představuje pravděpodobnost, že dvojice *Variables* a *Value* je součástí tohoto klastru. Při najetí myší na graf se zobrazí pravděpodobnost v procentech. Když se zde zobrazí hodnota nula, tak to neznamená, že zde nemusí tato hodnota patřit ale, že je zvýhodněn druhý klastr.
  - **Favorst <Cluster 2>** – sloupcový graf představuje pravděpodobnost, že dvojice *Variables* a *Value* je součástí tohoto klastru. Při najetí myší na graf se zobrazí pravděpodobnost v procentech. Když se zde zobrazí hodnota nula, tak to neznamená, že zde nemusí tato hodnota patřit ale, že je zvýhodněn první klastr.

6.1.3 Microsoft Decision Trees Algorithm

Zobrazený model, který je sestaven pomocí Microsoft Clustering Algorithm obsahuje tyto záložky *Decision Tree* a *Dependency Network*.

Záložka Decision Tree
Umožnuje prozkoumávání rozhodovacího stromu, které model obsahuje. Popis rozhraní záložky *Decision Tree* je následující, ve které se nacházejí tyto prvky:

- **Histograms:**
  Nastavení počtu zobrazených stavů v histogramu pro každý uzel. Pokud je počet stavů menší než je zvolená hodnota, tak se již nezobrazí další stavy v histogramu.

- **Tree:**
  Zvolení rozhodovacího stromu, který si chceme zobrazit. Počet rozhodovacích stromů závisí na počtu odhadovaných atributů v modelu.

- **Background:**
  Vybrání hodnoty, které dosahuje odhadovaný atribut, přičemž podle této hodnoty se obarví pozadí každého uzlu. Čím tmavší uzel je, tím větší podíl má na vybrané hodnotě.

- **Default Expansion:**
  Nastavení výchozí hodnoty zobrazení počtu úrovní rozhodovacího stromu.

- **Show Level:**
  Nastavení zobrazení počtu úrovní, které obsahuje rozhodovací strom. Nastavením na nejvyšší hodnotu se odkryjí všechny uzly v rozhodovacím stromu.

Záložka Dependency Network
Zobrazuje grafický pohled na všechny atributy, které model pro dolování dat obsahuje. Ukazuje, jak jsou jednotlivé atributy propojené. Tato záložka obsahuje intuitivní ovládací prvky, které popisovat nebudeme. Posuvníkem, který se nachází v ovládacích prvcích, můžeme postupně všechny hrany odstranit, nebo si je naopak nechat všechny zobrazit. Hrany představují prediktivní sílu propojení mezi odhadovaným a vstupním atributem. Ty hrany, které se při posouvání odstraňují, reprezentují menší důležitost hran mezi odhadovaným a vstupním atributem.

6.1.4 Microsoft Neural Network Algorithm

Zobrazený model, který je sestaven pomocí Microsoft Neural Network Algorithm neobsahuje žádné záložky, ale vše se zobrazuje v jednom okně. Popis tohoto rozhraní je následující:

- **Inputs:**
  Slouží ke zvolení atributu a hodnoty, které tento model používá jako vstupy. Ve výchozím nastavení se prohlížení modelu otevře se všemi atributy, které model obsahuje. U atributu zvolí hodnoty, jaké jsou nejdůležitější pro toto zobrazení. Zvolit atribut smíme ve sloupci *Attribute*, přičemž hodnotu tohoto atributu zvolíme ve sloupci *Value*. Postupně můžeme zvolit libovolné množství zkoumaných vstupních atributů a jejich hodnot.

- **Outputs:**
  Slouží ke zvolení odhadovaného atributu, který chceme prozkoumat. Dále si můžeme zvolit dva stavy, které chceme porovnat. Toto porovnání se provede v panelu *Variables*, kde zadáváme první (*Value 1*) a druhou (*Value 2*) porovnávanou hodnotu.

- **Variables:**
  Tato část obsahuje interaktivní sloupcové grafy, které reagují na konfiguraci v částech *Inputs* a *Outputs*. Neuronová sít vypočítává pravděpodobnost, že konkrétní hodnota ovlivňuje konkrétní výsledek. Vše se nám zobrazuje v tabulce, která má tyto sloupce:
  - **Attribute** – atribut.
  - **Value** – hodnota atributu.
  - **Favors 0** – zobrazení jak moc je ovlivněna první (*Value 1*) porovnávaná hodnota na základě vstupních atributů. Při najetí myši na konkrétní záznam se následně zobrazí informace o skóre a o procentuálním porovnání vůči *Favors 1*.
  - **Favors 1** – zobrazení jak moc je ovlivněna druhá (*Value 2*) porovnávaná hodnota na základě vstupních atributů. Při najetí myši na konkrétní záznam se následně zobrazí informace o skóre a o procentuálním porovnání vůči *Favors 0*.

6.1.5 Microsoft Sequence Clustering Algorithm

Zobrazený model, který je sestaven pomocí Microsoft Sequence Clustering Algorithm obsahuje tyto záložky *Cluster Diagram*, *Cluster Profiles*, *Cluster Characteristics*, *Cluster Discrimination* a *State Transition*.

Záložka Cluster Diagram
Poskytuje grafický přehled všech klastrů, které tento model obsahuje. Tato záložka poskytuje intuitivní ovládací prvky, které popisovat nebudeme. Jednotlivé klastry můžeme přejmenovávat. Při najetí myší na klastr se nám zobrazí počet prvků nacházející se v klastru. Popíšeme si pouze tyto nejdůležitější prvky:

- **Density:**
  Odstínové zobrazení počtu prvků v klastru. O jaký prvek či prvky se jedná, závisí na nastaveném parametru *Shading Variable*.

- **Links:**
  Slouží k nastavení zobrazení nejsilnějších vztahů mezi jednotlivými klastry. Snížením se ponechávají pouze nejsilnější vazby mezi klastry.

- **Shading Variable:**
  Výběr atributu, který je zastoupen v jednotlivých klastrech popřípadě zvolení celé populace (*Population*).

- **State:**
  Zvolení jednoho stavu, který se použije pro stínování v grafu.

Záložka Cluster Profiles
Poskytuje celkový přehled klastrů, které tento model obsahuje. Zobrazují se zde všechny atributy, společně s barevným rozlišením sekvencí v každém klastru. Popis rozhraní záložky *Cluster Profiles* je následující, ve které se nacházejí tyto prvky:

- **Show Legend:**
  Ukazuje sekvence (stavy) v klastru, které jsou zobrazeny barevně a následně textově.

- **Histogram Bars:**
  Nastavení zobrazení počtu stavů v histogramu. Pokud existuje více stavů než je nastavený počet, tak se zobrazí pouze stavy, které mají nejvyšší pravděpodobnost.

- **Attributes and Cluster Profiles:**
  Zobrazení seznamu sekvencí, které byly nalezeny. Každý klastr zobrazuje počet sekvencí, který je nastaven pomocí *Histogram Bars*. Nachází se zde dvě sady histogramů, kdy každý je na jiném řádku. V řádku kde se nachází jméno sekvenčního atributu s příponou „*Sample*“, se nachází seznam sekvencí pro každý klastr. V řádku kde se nachází pouze jméno sekvenčního atributu, se nachází zastoupení všech položek, které klastr obsahuje a jejich celkového rozložení. Pokud zvolíme nějaký klastr a řádek, zobrazí se detailnější výpis v legendě.

- **States:**
  Poskytuje informace o barvách každého stavu v řádku klastru.

Záložka Cluster Characteristics
Umožnuje prozkoumávání jednotlivých klastrů, které model obsahuje. Popis rozhraní záložky *Cluster Characteristics* je následující, ve které se nacházejí tyto prvky:

- **Cluster:**
  Zvolení klastru, který chceme zobrazit. Další možností je zvolení „*Population (All)*“, kde uvidíme rozložení sekvencí v celém modelu jako celku.

- **Characteristics for <cluster>:**
  Zobrazuje seznam sekvencí, které byly přiřazeny k aktuálně zvolenému klastru. Obsahuje následující sloupce, ve kterých jsou obsaženy informace zvoleného klastru:
  - **Variables** – tento sloupec udává, zda se jedná o hodnotu či přechod. Když se jedná pouze o hodnotu, obsahuje název atributu. Pokud se ale jedná o přechod, tak obsahuje název atributu s příponou „*Transitions*“.
  - **Values** – hodnota tohoto slupce závisí na tom, zda se jedná o hodnotu či přechod. Když se jedná o hodnotu, sloupec obsahuje hodnotu stavu. Pokud se jedná o přechod, sloupec obsahuje dvě hodnoty odděleny čárkou. Popřípadě je napsán stav přechodu, například jedná-li se o počátek sekvence je označena „[Start]->“ a pak následuje počáteční stav.
  - **Probability** – tento sloupec udává relativní pravděpodobnost, že tato hodnota popřípadě sekvence patří do zvoleného klastru.

Záložka Cluster Discrimination
Poskytuje možnost porovnat dva klastry, které obsahuje model. Popis rozhraní záložky *Cluster Discrimination* je následující, ve které se nacházejí tyto prvky:

- **Cluster 1:**
  Zvolení klastru, který chceme porovnat s jiným klastrem.

- **Cluster 2:**
  Zvolení druhého klastru, se kterým se provede porovnání k prvnímu zvolenému klastru. Porovnat také smíme doplněk k prvnímu klastru, což zobrazí všechny případy, které nejsou v *Cluster* 1.

- **Discrimination scores for <cluster 1> and <cluster 2>:**
  Zobrazení detailního srovnání klastrů, které jsme vybrali. Obsahuje následující sloupce, ve kterých jsou obsaženy informace o porovnávaných klastrech:
  - **Variables** – tento sloupec udává, zda se jedná o hodnotu či přechod. Když se jedná pouze o hodnotu, obsahuje název atributu. Pokud se ale jedná o přechod, tak obsahuje název atributu s příponou „*Transitions*“.
  - **Values** – hodnota tohoto slupce závisí na tom, zda se jedná o hodnotu či přechod. Když se jedná o hodnotu, sloupec obsahuje hodnotu stavu. Pokud se jedná o přechod, hodnoty jsou odděleny těmito znaky „->“. Popřípadě je napsán stav

přechodu, například jedná-li se o počátek sekvence je označena „[Start]->“ a pak následuje počáteční stav.

- **Favors <Cluster 1>** – sloupcový graf představuje pravděpodobnost, že hodnota či sekvence je součástí tohoto klastru. Při najetí myší na graf se zobrazí pravděpodobnost v procentech. Když se zde zobrazí hodnota nula, tak to neznamená, že zde nemusí tato hodnota patřit ale, že je zvýhodněn druhý klastr.
- **Favorst <Cluster 2>** – sloupcový graf představuje pravděpodobnost, že hodnota či sekvence je součástí tohoto klastru. Při najetí myší na graf se zobrazí pravděpodobnost v procentech. Když se zde zobrazí hodnota nula, tak to neznamená, že zde nemusí tato hodnota patřit ale, že je zvýhodněn první klastr.

Záložka State Transitions

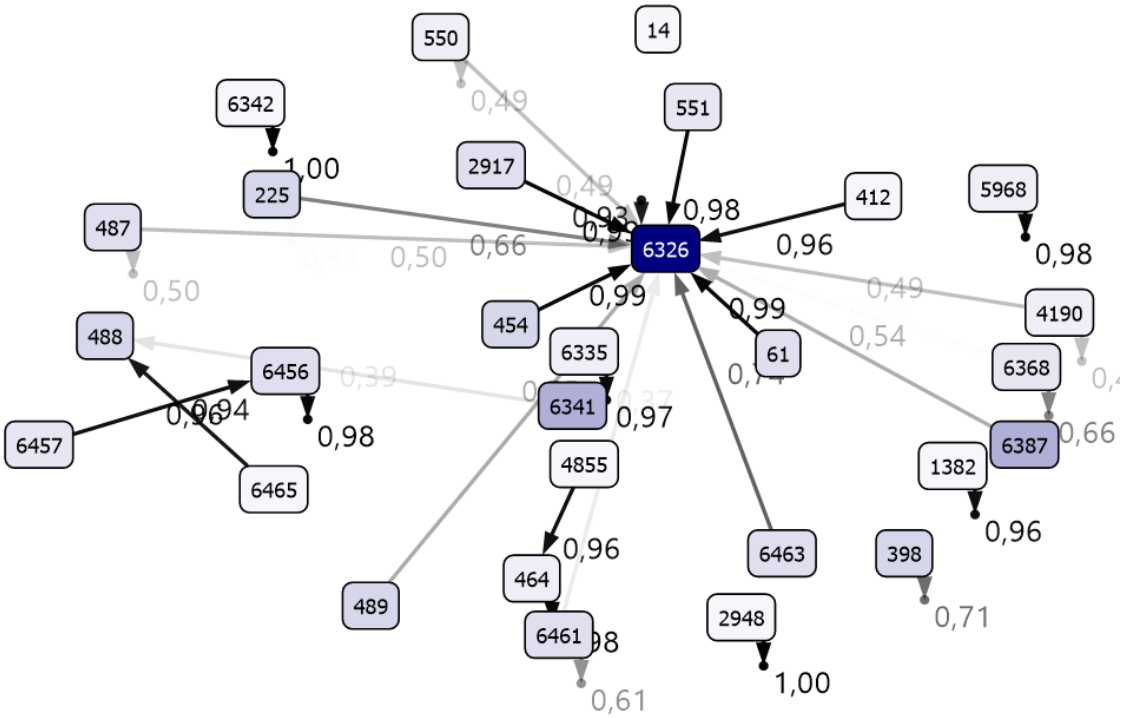
V této záložce můžeme vidět v grafické podobě jednotlivé přechody mezi hodnotami, které jsou ve zvoleném klastru. Tato záložka poskytuje intuitivní ovládací prvky, které popisovat nebudeme. Popis rozhraní záložky *State Transitions* je následující, ve které se nacházejí tyto prvky:

- **Cluster:**
  Zvolení klastru, který si chceme zobrazit. Další možností je zvolení „*Population (All)*“, kde uvidíme rozložení sekvencí v celém modelu jako celku.

- **Show Edge Labels:**
  Zobrazení pravděpodobnosti přechodu, která se zobrazí u každé hrany v grafu.

- **Links:**
  Slouží k nastavení zobrazení nejpravděpodobnějších přechodů mezi jednotlivými stavy v klastru. Snížením se ponechávají pouze nejsilnější přechody mezi stavy v klastru.

Na obrázku 26 můžeme vidět příklad zobrazení grafu pro model sestavený pomocí Microsoft Sequence Clustering Algorithm. Jak vyčíst informace o přechodech si nyní popíšeme. Když se budeme nacházet v uzlu 551 tak s pravděpodobností 98% přejdeme do uzlu 6326. Počáteční stav poznáme tak že, že se u něj nachází šipka s tečkou na začátku, jako je tomu například u uzlu 6326. Koncový stav poznáme tak, že se u něj nachází šipka s tečkou ve špičce, jako je tomu například u uzlu 550, kdy s 49% pravděpodobností v tomto stavu zůstaneme.



Obrázek 26: Graf přechodů vytvořený pomocí Microsoft Sequence Clustering Algorithm

6.2 Přesnost modelů pro dolování dat

V Microsoft Visual Studio 2012 se nachází možnost zobrazení přesnosti modelů pro dolování dat. Otestování lze provést pro modely, které mají alespoň jeden odhadovaný atribut. Modelem myslíme tím, model pro dolování dat, který je již vázán na některý z algoritmů pro dolování dat a je trénován na konkrétních datech. Výpočet přesnosti modelů se provádí ve struktuře pod záložkou *Mining Accuracy Chart*. Tento výpočet se spouští automaticky při zobrazení grafu přesnosti modelu, za podmínky, že se v testovací množině nacházejí nějaká data. Aby testovací množina obsahovala nějaká data, musí být některý z parametrů *HoldoutMaxCases* nebo *HoldoutMaxPercent* nastaven na nenulovou hodnotu. Testovaná data se mohou použít buď, ze stejné tabulky jako byl model pro dolování dat trénován, nebo zvolíme jinou tabulku. Při zvolení jiné tabulky se v testovací množině nemusí nacházet žádná data. Graf přesnosti se nachází v podzáložce *Lift Chart*. Určení přesnosti modelů pro dolování dat můžeme provést také manuálně, kdy si provedeme odhady nových případů. Odhad nových případů můžeme pro naše testovaná data provést, jelikož známe jejich výsledky a víme, zda se jedná o síťový útok či nikoliv.

6.2.1 Mining Accuracy Chart

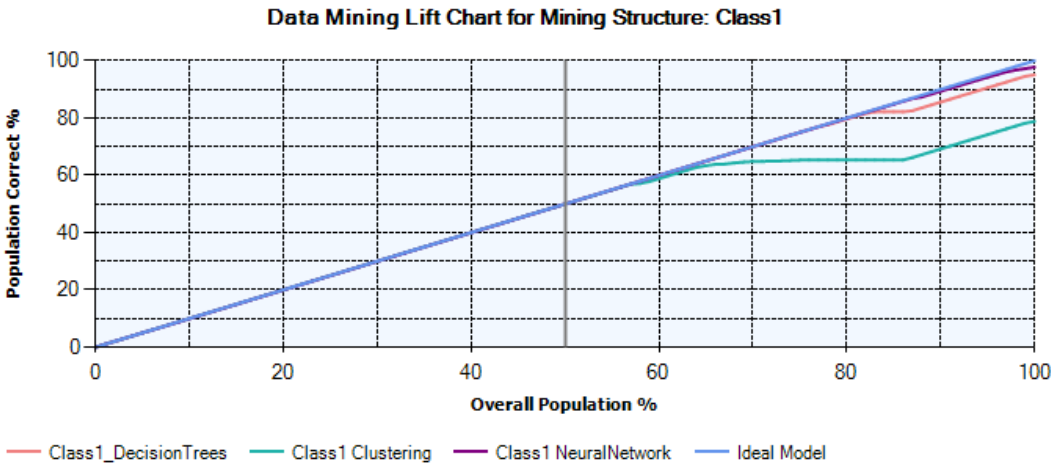
V záložce *Mining Accurary Chart* se nachází několik dalších záložek. Obsahuje celkem další čtyři záložky, přičemž pro nás jsou nejdůležitější první dvě, které si popíšeme. Ostatní jsou popsány v dokumentaci [2].

Záložka Input Selection
V této záložce můžeme vidět modely, jež obsahuje struktura. V této záložce můžeme vidět a definovat:

- Modely, které se nacházejí ve struktuře. Současně se mohou otestovat všechny modely nebo námi zvolené. Pokud model odhaduje více atributů, nastavíme pak, pro který odhadovaný atribut se má testování provést, popřípadě pro jakou odhadovanou hodnotu.

- Výběr dat, na kterých se má provést testování sestaveného modelu. Na výběr máme z několika možností, kdy jejich podrobný popis najdeme v dokumentaci [2]. Ve výchozím nastavení se model otestuje nad daty, z kterých se trénoval. Nebo můžeme zvolit tabulku s daty, pro která se má provést testování. Jedná se o volbu *Specify a different data set*. Na výběr pak máme tabulky a pohledy které jsou namapovány v projektu (kapitola 5.1.1).

Záložka Lift Chart
Zobrazuje vypočtený graf úspěšnosti odhadů pro otestovaný model pro dolování dat. V tomto grafu se zobrazuje ideální křivka, označená modrou barvou, kterou by měly ostatní modely pro dolování dat kopírovat. Křivka pro správně navržený model pro dolování dat by měla co nejvíce odpovídat ideální křivce a její sklon by se měl kolem ní pohybovat. Pokud tomu tak není, může se jednat o nesprávně zvolený algoritmus pro dolování dat či špatně nastavený model pro dolování dat. Ukázku tohoto grafu můžeme vidět na obrázku 27. Můžeme si také zobrazit legendu tohoto grafu, ve které se nachází informace o úspěšnosti modelů.



Obrázek 27: Grafická ukázka otestování úspěšnosti modelů pro dolování dat

6.3 Vytváření nových odhadů

Vytváření nových odhadů budeme provádět v záložce *Mining Model Prediction*, Nejprve musíme zvolit vstupní tabulku, u které chceme odhadovat výsledky. Klikneme na tlačítko „*Select Case Table*...“, následně se nám otevře okno, ve kterém zvolíme tabulku *testcl1*. Pro vytváření odhadu slouží spodní část obrazovky, kdy nastavování provádíme v jednotlivých sloupcích.

Postup jak vytvářet nové odhady si popíšeme v následujících krocích na ukázkovém příkladu:

1. Ve sloupci *Source* nastavíme postupně tyto vlastnosti:
   a. Vybereme vstupní tabulku (*testcl1*), u které budeme odhadovat výsledky,
   b. zvolíme si model (*Class1_DecisionTrees*) pro odhadování výsledků atributů,
   c. vybereme odhadovací funkci (*Prediction Function*).

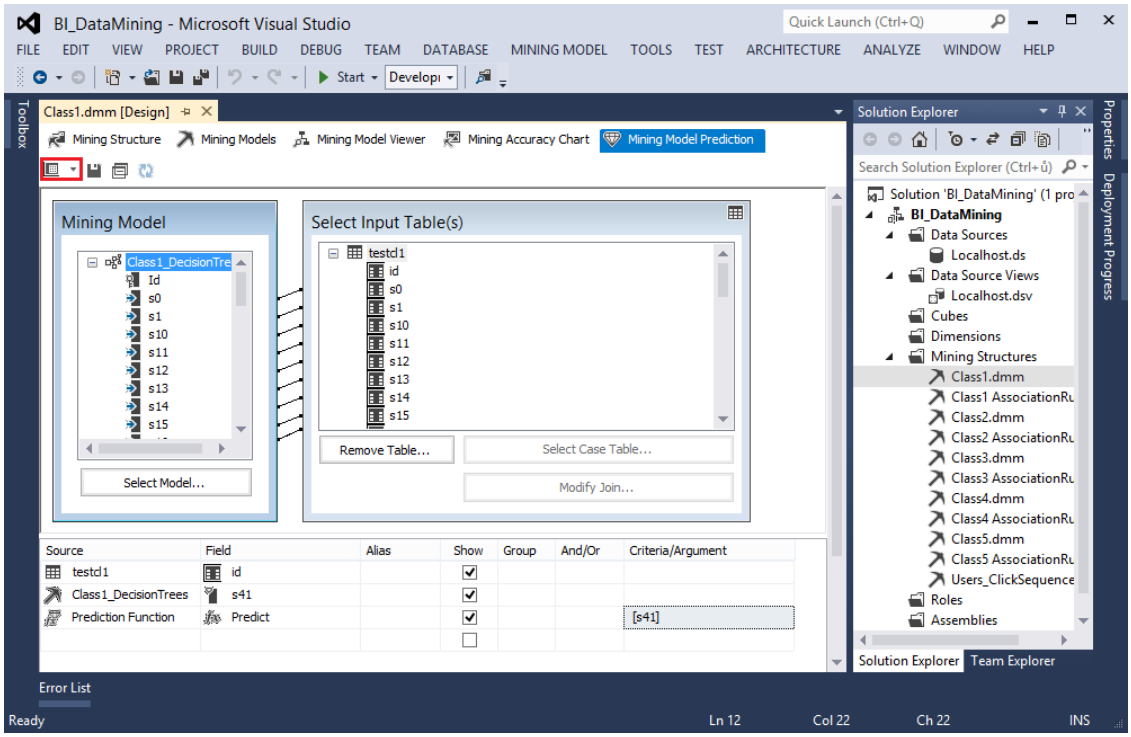
2. Ve sloupci *Field* máme nyní klíčový sloupec (*id*) zvolené tabulky a také zde máme odhadovaný atribut (*s41*). Pokud model obsahuje více odhadovaných atributů, můžeme si vybrat atribut, se kterým bude model pracovat. Na výběr máme z několika možností jako je například provedení odhadu, výpočet pravděpodobnosti, přiřazení do klastru atd. Výčet všech možností včetně jejich popisů najdeme v dokumentaci [2]. My zvolíme možnost *Predict* (odhadování).

3. Ve sloupci *Alias* si můžeme jednotlivé sloupce pojmenovávat. Pokud název nezadáme, použije se název ze sloupce *Field*.

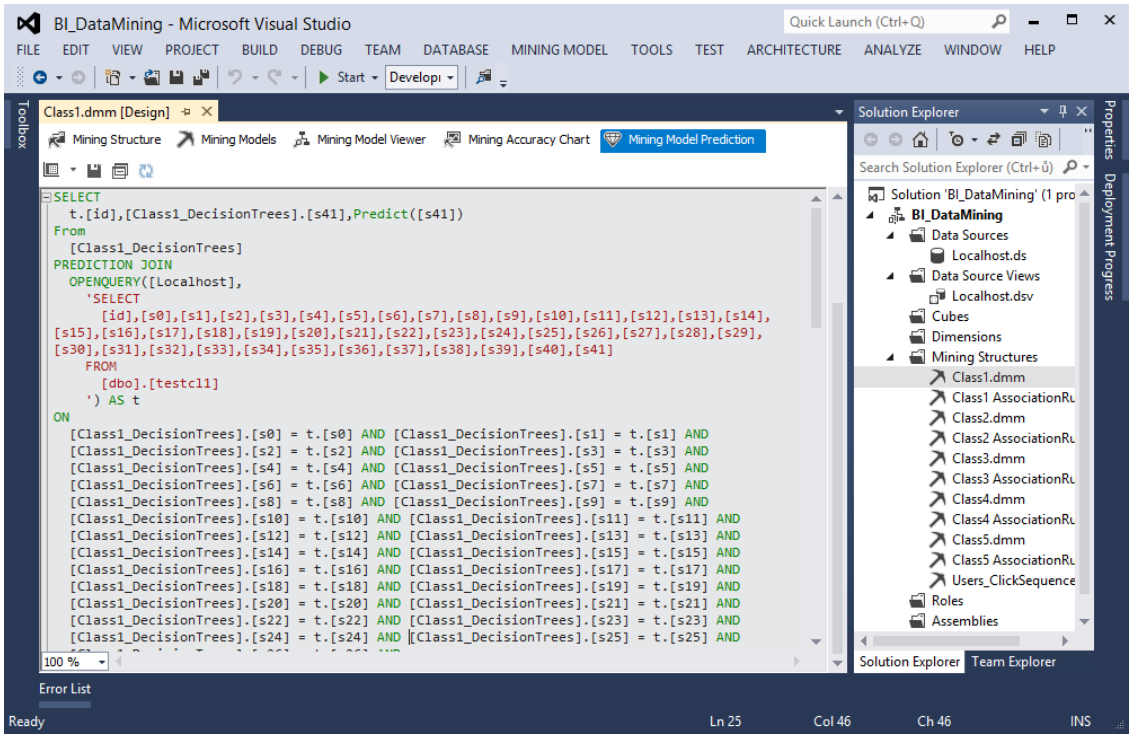
4. Ve sloupci „*Criteria/Argument*“ pro řádek ve kterém se nachází *Predict Funcition*, musíme vyplnit „[s41]“ tímto nastavujeme, který atribut chceme odhadovat. Jakmile zadáme špatný atribut, odhad se neprovede, ale zobrazí se nám chyba. Nyní nám stačí kliknout v horní části na tlačítko v červeném rámečku na obrázku 28 a zvolit možnost *Result*.

Pro celý tento postup se nám poté vygeneruje SQL příkaz, který si můžeme zobrazit při kliknutí na šipku na obrázku 28 v červeném rámečku a zvolit možnost *Query*. Poté se nám v dolní části zobrazí vygenerovaný SQL příkaz odhadovaného dotazu. Kolik práce jsme ušetřili interaktivním vyplněním, si můžeme prohlédnout na obrázku 29. Odhady můžeme psát taky pomocí SQL dotazů přímo v zobrazeném okně.

Prohlížený odhad si můžeme uložit přímo do tabulky databáze. To provedeme tak, že klikneme na tlačítko *Save Query Result*. Zobrazí se nám dialog, ve kterém můžeme uložit výsledek odhadu či jiné funkce kterou jsme zvolili, do tabulky databáze s námi zadaným názvem. Když tabulka s tímto názvem neexistuje, tak se vytvoří. Pokud existuje, budeme o tom informování a můžeme případně uložit výsledek odhadu do této tabulky.



Obrázek 28: Ukázka nastavení pro provedení odhadu



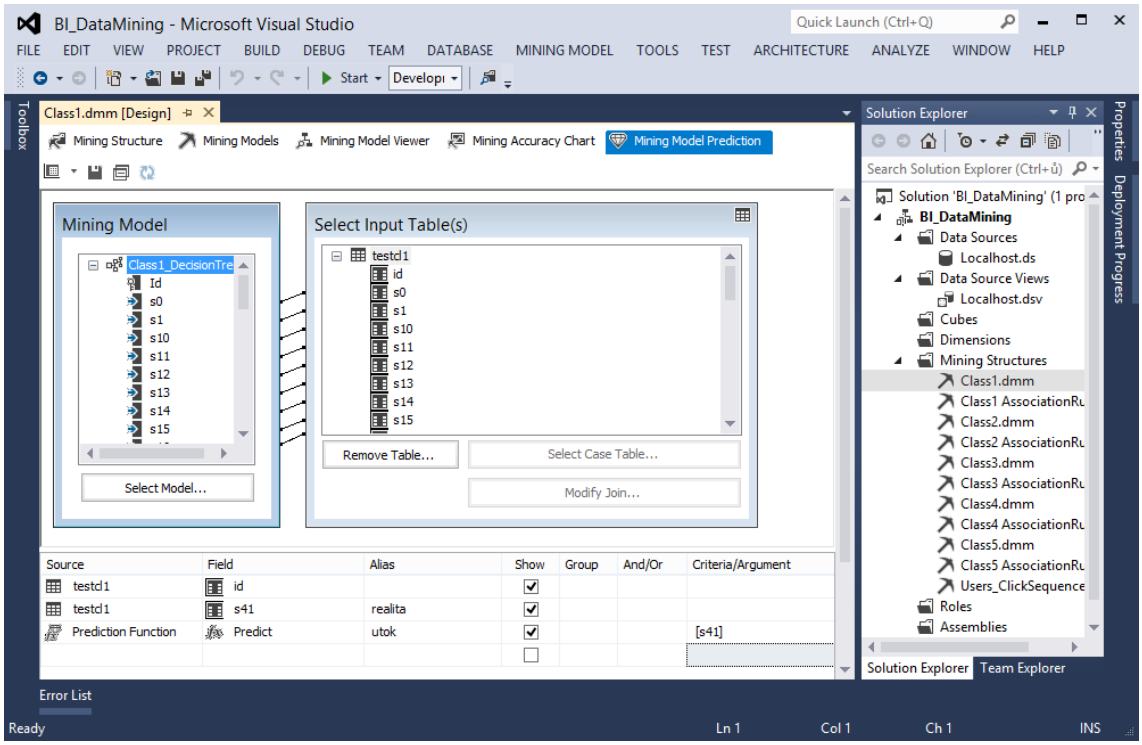
Obrázek 29: Ukázka vygenerovaného SQL dotazu pro provedení odhadu

6.4 Analýza úspěšnosti odhadů vytvořených modelů

Nyní provedeme analýzu úspěšnosti odhadů jednotlivých modelů na experimentálních datech. Porovnáme úspěšnost základního modelu, ve kterém jsou všechny parametry u nastaveného algoritmu ve výchozím nastavení. V druhém případě poupravíme některé parametry u algoritmů pro dolování dat se snahou dosáhnout lepších výsledků. Testování budeme provádět s daty, ve kterých se nacházejí síťové útoky, neboť předem známe jejich výsledek.

Jak lze vytvářet nové odhady jsme si ukázali v kapitole 6.3. Provedeme drobnou úpravu, abychom ve vytvořeném odhadu měli informaci o reálné hodnotě síťového útoku. Tuto úpravu můžeme vidět na obrázku 30. Atribut *realita* ponese reálnou informaci o tom, zda se jedná o síťový útok či nikoliv. Atribut *utok* ponese odhadnutou informaci, zda se jedná o síťový útok či nikoliv.

Pro analýzu úspěšnosti odhadu budeme využívat SQL dotazy, které se nacházejí v SQL skriptu „*AnalyzaUspesnostiOdhadu.sql*“. Skript byl vytvořen pro naší analýzu úspěšnosti odhadů, nachází se na DVD médiu, které je součástí přílohy této práce. Tento uložený skript budeme spouštět v Microsoft SQL Server Management Studio. V tomto skriptu stačí pouze zaměnit název tabulky v proměnné „*NazevTabulky*“, kde zadáme název tabulky, do které jsme uložili výsledek odhadu s názvem atributů (*realita*, *utok*), jako je tomu na obrázku 30.



Obrázek 30: Ukázka nastavení pro provedení odhadu

U všech modelů pro dolování dat byl následně upraven parametr *HoldoutMaxPercent* na hodnotu 0. Tímto se modely pro dolování dat budou trénovat ze všech vstupních dat. Další parametry, které byly upraveny u jednotlivých modelů pro dolování dat, jsou uvedeny v tabulce 8. Výsledky úspěšnosti měření se nachází v tabulkách 9, 10, 11, 12 a 13. V názvu tabulek je uvedeno, jaká dvojice byla testována. Nejprve je uvedena tabulka, ze které se model trénoval a poté je uveden název tabulky, pro kterou se prováděl odhad.

Tabulka 8: Přehled upravených parametrů jednotlivých modelů

Model	Parametr	Nastavená hodnota
Class1		
AssociationRules	Minimum probability Minimum support	0,99 0,07
Clustering	Cluster seed	5
DecisionTrees	Score method	1
NeuralNetwork	Holdout percentage Maximum states	10 10
Class2		
AssociationRules	Minimum probability Minimum support	0,95 0,04
Clustering	Cluster seed	5
DecisionTrees	Split method	1
NeuralNetwork	Holdout percentage Holdout seed	40 5
Class3		
AssociationRules	Minimum probability Minimum support	0,99 0,05
Clustering	Cluster seed	10
DecisionTrees	Split method	2
NeuralNetwork	Holdout percentage Holdout seed	10 20
Class4		
AssociationRules	Minimum support	0,0005
DecisionTrees	Minimum support	15
NeuralNetwork	Holdout percentage Holdout seed	50 20
Class5		
AssociationRules	Minimum probability	0,85
Clustering	Cluster seed	4
DecisionTrees	Score method Split method	3 1
NeuralNetwork	Holdout percentage Holdout seed	50 100

Tabulka 9: Analýza úspěšnosti modelů v základním nastavení pro class1 - testcl1

	AssociationRules	**Clustering**	**DecisionTrees**	**NeuralNetwork**
Celkový počet dat	6890			
Reálný počet útoků	1400			
Celkový počet odhadnutých útoků	1837	2281	2711	2062
Správně odhalené útoky	1396 99,71%	1269 90,64%	**1397** **99,79%**	1396 99,71%
Neodhalené útoky	4 0,29%	131 9,36%	**3** **0,21%**	4 0,29%
Celková chyba odhadů	**445** **6,46%**	1143 16,59%	1317 19,11%	670 9,72%

Tabulka 10: Analýza úspěšnosti modelů s upravenými parametry pro class1 - testcl1

	AssociationRules	**Clustering**	**DecisionTrees**	**NeuralNetwork**
Celkový počet dat	6890			
Reálný počet útoků	1400			
Celkový počet odhadnutých útoků	1594	2129	1163	1493
Správně odhalené útoky	**1396** **99,71%**	1287 91,93%	1112 79,43%	**1396** **99,71%**
Neodhalené útoky	**4** **0,29%**	113 8,07%	288 20,57%	**4** **0,29%**
Celková chyba odhadů	202 2,93%	955 13,86%	339 4,92%	**101** **1,47%**

Pro testovací kolekci *Class1* byl nejlepším modelem *NeuralNetwork* s upravenými parametry. Dosáhnul nejlepších výsledků v celkové chybovosti v odhadech a to 1,47% (101 nesprávných odhadů). Tento model stejně jako *AssociationRules* s upravenými parametry je nejlepším modelem se správně odhalenými útoky. Kdy správně odhalily útoky v 99,71% (1396 správných odhadů).

Celkově nejhorším modelem s upravenými parametry pro testovací kolekci *Class1* je *Clustering*. Dosáhnul nejhorších výsledků v celkové chybovosti v odhadech a to 13,86% (955 nesprávných odhadů). Nejméně správných útoků odhalil model s upravenými parametry *DecisionTrees*, přitom nedosahuje největší chybovosti v odhadu.

Úpravou parametrů došlo k výraznému zlepšení vůči celkové chybovosti v odhadu u modelu *DecisionTrees*. Došlo však také k výraznému zhoršení ve správnosti odhalených útoků. Naopak model *Clustering* si polepšil nejméně.

Tabulka 11: Analýza úspěšnosti modelů v základním nastavení pro class2 – testcl2

	AssociationRules	**Clustering**	**DecisionTrees**	**NeuralNetwork**
Celkový počet dat	6890			
Reálný počet útoků	700			
Celkový počet odhadnutých útoků	563	441	979	1273
Správně odhalené útoky	397 56,71%	356 50,57%	694 99,14%	**698** **99,71%**
Neodhalené útoky	303 43,29%	346 49,43%	6 0,86%	**2** **0,29%**
Celková chyba odhadů	469 6,81%	433 6,28%	**291** **4,22%**	577 8,37%

Tabulka 12: Analýza úspěšnosti modelů s upravenými parametry pro class2 – testcl2

	AssociationRules	**Clustering**	**DecisionTrees**	**NeuralNetwork**
Celkový počet dat	6890			
Reálný počet útoků	700			
Celkový počet odhadnutých útoků	426	379	715	830
Správně odhalené útoky	396 56,57%	379 54,14%	697 99,57%	**700** **100%**
Neodhalené útoky	304 43,43%	321 45,86%	3 0,43%	**0** **0%**
Celková chyba odhadů	334 4,85%	321 4,66%	**21** **0,30%**	130 1,89%

Pro testovací kolekci *Class2* byl nejlepším modelem *DecisionTrees* s upravenými parametry. Dosáhnul nejlepších výsledků v celkové chybovosti v odhadech a to 0,30% (21 nesprávných odhadů). Kdy správně odhalil útoky v 99,57% (697 správných odhadů). Nejlepším modelem se správně odhalenými útoky je *NeuralNetwork* s upravenými parametry s výsledkem 100% (700 správných odhadů). Jeho celková chybovost v odhadu je 1,89% (130 nesprávných odhadů), protože odhadoval útok u dalších 130 záznamů.

Celkově nejhorším modelem s upravenými parametry pro testovací kolekci *Class2* je *AssociationRules*. Dosáhnul nejhorších výsledků v celkové chybovosti v odhadech a to 4,85% (334 nesprávných odhadů). Model *Clustering* s upravenými parametry odhalil nejméně správných útoků při experimentech na datech, přitom nedosahuje největší chybovosti v odhadu.

Úpravou parametrů došlo k výraznému zlepšení vůči celkové chybovosti v odhadu u modelu *NeuralNetwork*. Naopak model *Clustering* si polepšil nejméně.

Tabulka 13: Analýza úspěšnosti modelů v základním nastavení pro class3 – testcl3

	AssociationRules	**Clustering**	**DecisionTrees**	**NeuralNetwork**
Celkový počet dat	6890			
Reálný počet útoků	4202			
Celkový počet odhadnutých útoků	2488	3154	4098	3672
Správně odhalené útoky	2488 59,21%	3074 73,16%	**3994** **95,05%**	3524 83,86%
Neodhalené útoky	1714 40,79%	1128 26,84%	**208** **4,95%**	678 16,14%
Celková chyba odhadů	1714 24,88%	1208 17,53%	**312** **4,53%**	826 11,99%

Tabulka 14: Analýza úspěšnosti modelů s upravenými parametry pro class3 – testcl3

	AssociationRules	**Clustering**	**DecisionTrees**	**NeuralNetwork**
Celkový počet dat	6890			
Reálný počet útoků	4202			
Celkový počet odhadnutých útoků	4216	4712	4140	4149
Správně odhalené útoky	3901 92,84%	3879 92,31%	4040 96,14%	**4145** **98,64%**
Neodhalené útoky	301 7,16%	323 7,69%	162 3,86%	**57** **1,36%**
Celková chyba odhadů	616 8,94%	1156 16,78%	262 3,80%	**61** **0,89%**

Pro testovací kolekci *Class3* byl nejlepším modelem *NeuralNetwork* s upravenými parametry. Dosáhnul nejlepších výsledků v celkové chybovosti v odhadech a to 0,89% (61 nesprávných odhadů). Kdy správně odhalil útoky v 98,64% (4145 správných odhadů), proto je také nejlepším modelem se správně odhalenými útoky.

Celkově nejhorším modelem s upravenými parametry pro testovací kolekci *Class3* je *Clustering*. Dosáhnul nejhorších výsledků v celkové chybovosti v odhadech a to 16,78% (1156 nesprávných odhadů). Tento model také odhalil nejméně správných útoků při experimentování s daty.

Úpravou parametrů došlo k výraznému zlepšení vůči celkové chybovosti v odhadu u modelu *AssociationRules*. Naopak model *DecisionTrees* si polepšil nejméně.

Tabulka 15: Analýza úspěšnosti modelů v základním nastavení pro class4 – testcl4

	AssociationRules	**Clustering**	**DecisionTrees**	**NeuralNetwork**
Celkový počet dat	6890			
Reálný počet útoků	25			
Celkový počet odhadnutých útoků	0	0	0	9
Správně odhalené útoky	0 0%	0 0%	0 0%	**1** **4%**
Neodhalené útoky	25 100%	25 100%	25 100%	**24** **96%**
Celková chyba odhadů	25 0,36 %	25 0,36 %	**25** **0,36 %**	32 0,46%

Tabulka 16: Analýza úspěšnosti modelů s upravenými parametry pro class4 – testcl4

	AssociationRules	**Clustering**	**DecisionTrees**	**NeuralNetwork**
Celkový počet dat	6890			
Reálný počet útoků	25			
Celkový počet odhadnutých útoků	57	0	18	18
Správně odhalené útoky	**16** **64%**	0 0%	15 60%	**16** **64%**
Neodhalené útoky	**9** **36%**	25 100%	10 40%	**9** **36%**
Celková chyba odhadů	50 0,73%	25 0,36 %	13 0,19%	**11** **0,16%**

Pro testovací kolekci *Class4* byl nejlepším modelem *NeuralNetwork* s upravenými parametry. Dosáhnul nejlepších výsledků v celkové chybovosti v odhadech a to 0,16% (11 nesprávných odhadů). Tento model stejně jako *AssociationRules* s upravenými parametry je nejlepším modelem se správně odhalenými útoky. Kdy správně odhalily útoky v 64% (16 správných odhadů).

Model *AssociationRules* dopadnul při experimentování s daty nejhůře, i přesto, že dokázal odhadnout některé reálné útoky. V celkové chybovosti v odhadech dosáhnul nejhorších výsledků a to 0,73% (50 nesprávných odhadů). Žádný útok neodhalil model *Clustering* s upravenými parametry, přitom nedosahuje největší chybovosti v odhadu. U tohoto modelu se úpravou jednotlivých parametrů nedosáhlo zlepšení, proto není vůbec uveden v tabulce 8.

Úpravou parametrů došlo k výraznému zlepšení vůči celkové chybovosti v odhadu u modelu *NeuralNetwork*. Model *AssociationRules* sice dokázal již správně odhadnout reálné útoky, avšak úpravou parametrů došlo ke zhoršení jeho celkové chybovosti v odhadu.

Tabulka 17: Analýza úspěšnosti modelů v základním nastavení pro class5 – testcl5

	AssociationRules	**Clustering**	**DecisionTrees**	**NeuralNetwork**
Celkový počet dat	6890			
Reálný počet útoků	563			
Celkový počet odhadnutých útoků	561	162	541	26
Správně odhalené útoky	527 93,61%	161 28,60%	**536** **95,20%**	26 4,62%
Neodhalené útoky	36 6,39%	402 71,40%	**27** **4,80%**	537 95,38%
Celková chyba odhadů	70 1,02%	403 5,85%	**32** **0,46%**	537 7,79%

Tabulka 18: Analýza úspěšnosti modelů s upravenými parametry pro class5 – testcl5

	AssociationRules	**Clustering**	**DecisionTrees**	**NeuralNetwork**
Celkový počet dat	6890			
Reálný počet útoků	563			
Celkový počet odhadnutých útoků	532	692	556	557
Správně odhalené útoky	525 93,25%	506 89,88%	**551** **97,87%**	**551** **97,87%**
Neodhalené útoky	38 6,75%	57 10,12%	**12** **2,13%**	**12** **2,13%**
Celková chyba odhadů	45 0,65%	243 3,53%	**17** **0,25%**	18 0,26%

Pro testovací kolekci *Class5* byl nejlepším modelem *DecisionTrees* s upravenými parametry. Dosáhnul nejlepších výsledků v celkové chybovosti v odhadech a to 0,25% (17 nesprávných odhadů). Tento model stejně jako *NeuralNetwork* s upravenými parametry je nejlepším modelem se správně odhalenými útoky. Kdy správně odhalily útoky v 97,87% (551 správných odhadů).

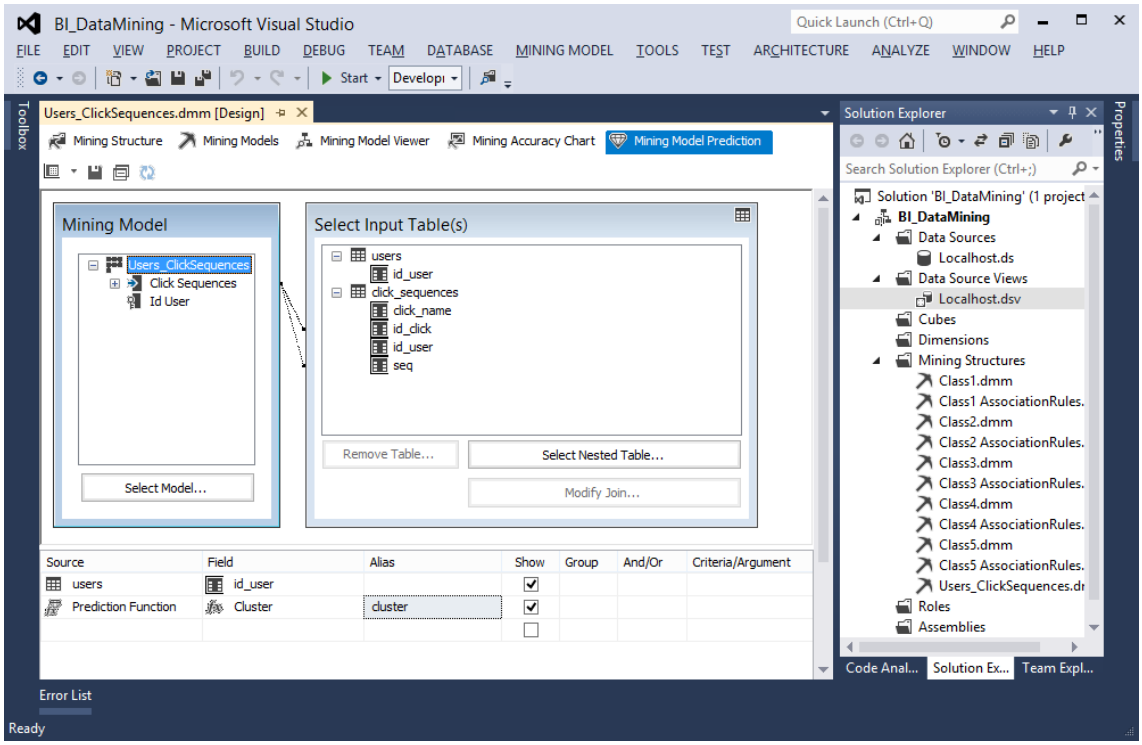
Celkově nejhorším modelem s upravenými parametry pro testovací kolekci *Class5* je *Clustering*. Dosáhnul nejhorších výsledků v celkové chybovosti v odhadech a to 3,53% (243 nesprávných odhadů). Tento model také odhalil nejméně správných útoků.

Úpravou parametrů došlo k výraznému zlepšení vůči celkové chybovosti v odhadu u modelu *NeuralNetwork*. Naopak model *DecisionTrees* si polepšil nejméně.

Správné odhalení útoků může být důležitější než samotná celková chyba v odhadu. A to z toho důvodu, že modely lépe identifikují opravdový síťový útok, přičemž odhadují více útoků, což může posloužit jako prevence, abychom mohli včas zasáhnout. Vždy bude záležet na datech a výsledcích, které se mají odhadovat.

6.5 Analýza modelu pro sekvenční data

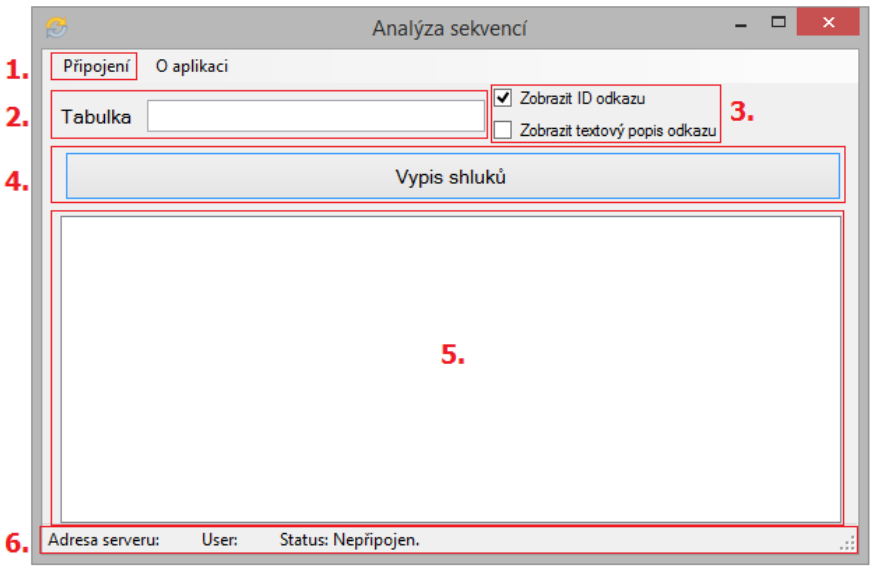
Nyní provedeme analýzu modelu, který zpracovává sekvenční data. Ukážeme si postup jak zjistit obsah jednotlivých klastrů, kdy si vypíšeme jednotlivé sekvence. Analýzu budeme provádět v modelu *Users_ClickSequences*. Nejprve musíme vytvořit odhad, ve kterém nastavíme, do jakého klastru patří sekvence. Toto nastavení můžeme vidět na obrázku 31.



Obrázek 31: Nastavení pro přiřazení sekvence do klastru

Poté klikneme na tlačítko *Result*, kde již uvidíme jedinečný identifikátor uživatele a do kterého klastru byl přiřazen. Tento výsledek opět uložíme do tabulky databáze kliknutím na tlačítko *Save query result*. Zobrazí se nám nové okno, ve kterém zadáme název tabulky *AnalyzaSekvence* a uložíme. Nyní použijeme aplikaci s názvem „*Analýza sekvencí*“.

Seznámíme se s tím, jak pracovat s aplikací pro analýzu sekvencí. Na obrázku 32 vidíme uživatelské rozhraní aplikace, včetně jejího popisu.



Obrázek 32: Popis uživatelského rozhraní aplikace pro analýzu sekvencí

Popis uživatelského rozhraní aplikace:

1. Kliknutím na toto tlačítko, nám vyskočí okno, ve kterém se budeme moci připojit k instanci databáze. Zadáváme:
    a. IP adresu,
    b. uživatelské jméno,
    c. heslo.

2. Zadání názvu tabulky.

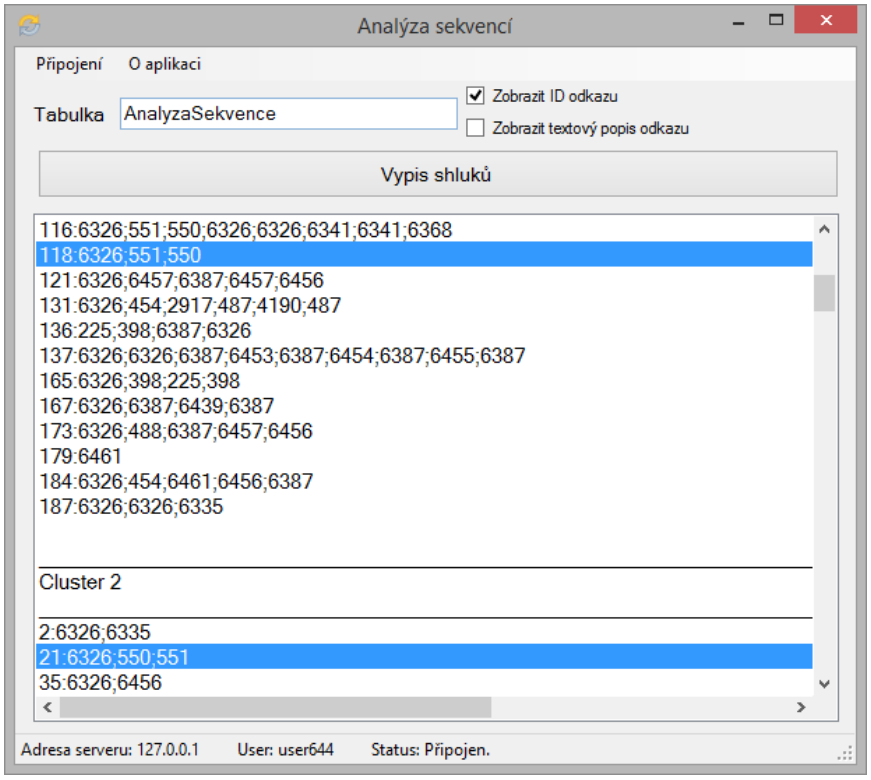
3. Nastavení zobrazení informací o sekvenci ve výpisu.

4. Výpis obsahu jednotlivých klastrů včetně sekvence.

5. Zobrazení klastrů a sekvencí.

6. Zde vidíme adresu databázového serveru, ke kterému se připojujeme, včetně uživatelského jména a stavu o tom zda jsme připojeni či ne.

Nejdříve si musíme vyplnit přihlašovací údaje k databázi v aplikaci pro analýzu sekvencí. Poté zadáme název tabulky *AnalyzaSekvence* a necháme si vypsat shluky včetně jejich obsahu. Ve výpisu uvidíme všechny klastry a jejich obsah včetně sekvencí. Na prvním místě se nachází jedinečný identifikátor uživatele. Za dvojtečkou následuje průchod stránkami, jak jej prováděl uživatel, přičemž oddělovačem jednotlivých stránek je znak středník. Vše můžeme vidět na obrázku 33. Na tomto obrázku se nám mohou zdát označené sekvence téměř stejné, ale každá je součástí jiného klastru. Sekvence *118:6326;551;550* se nachází v prvním klastru a sekvence *21:6326;550;551* se nachází v druhém klastru.

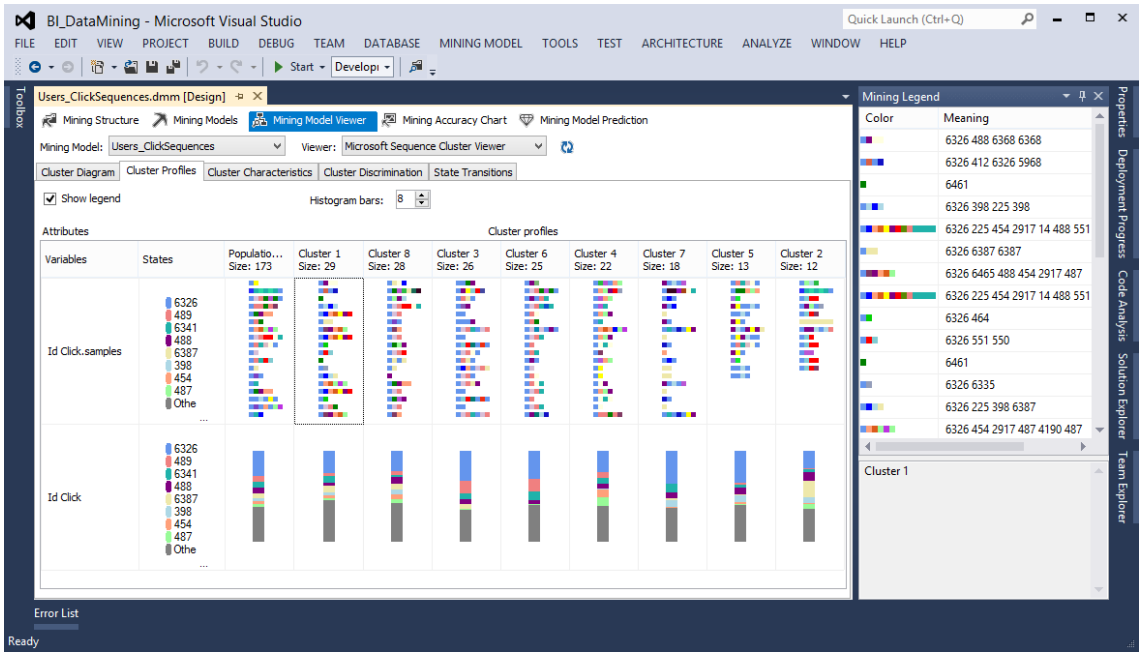


Obrázek 33: Zobrazení obsahů jednotlivých klastrů a přiřazených sekvencí

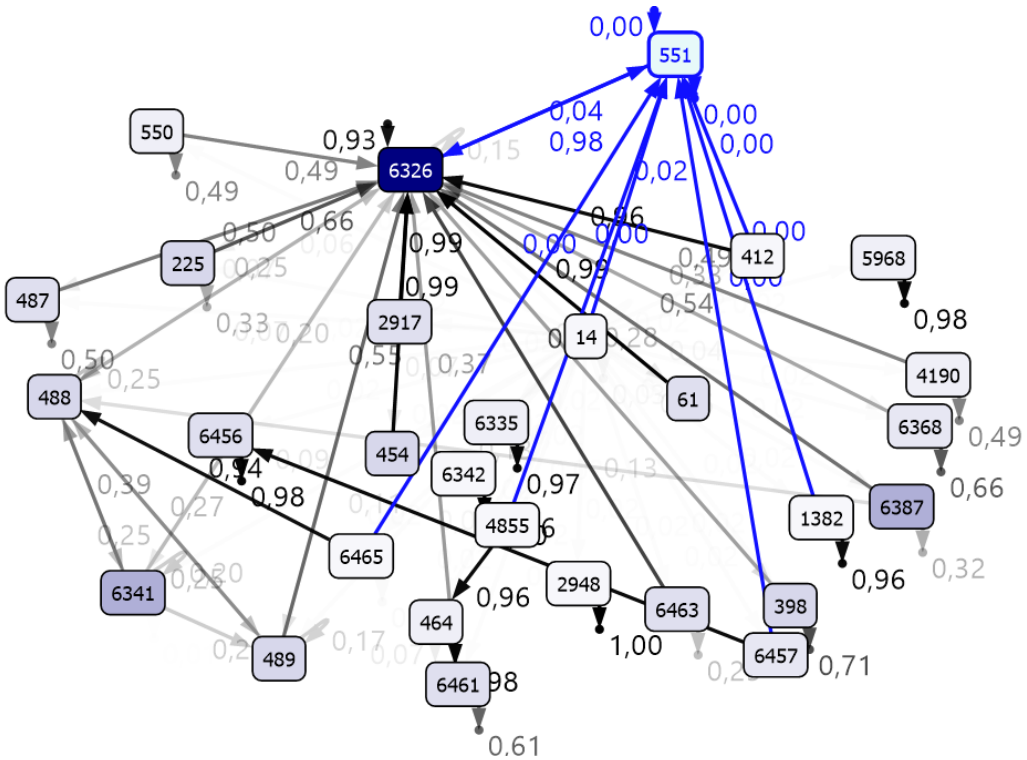
Nejdříve si můžeme prohlédnout obsah shluků jednotlivých klastrů v záložce *Cluster Profiles*. Na obrázku 34 můžeme vidět jednotlivé sekvence, které se nacházejí v každém shluku. Všimnout si například můžeme, kolik sekvencí obsahuje každý klastr (*Size*). Informace, které zde můžeme vyčíst, byly podrobněji uvedeny v kapitole 6.1.5.

Nejdříve si prohlédneme v záložce *State Transition*, která se nachází v *Mining Model Viewer* sestavený model *Users_ClickSequences* a důkladně jej prozkoumáme. Pokud si nejdříve zobrazíme *Cluster 1*, tak zjistíme, že s 93% pravděpodobností bude stav 6326 (stránka s jedinečným identifikátorem 6326) počátečním a existuje 4% pravděpodobnost, že budeme pokračovat do stavu 551. Dále pokud se nacházíme ve stavu 550, existuje 49% pravděpodobnost, že v tomto stavu zůstaneme, je tedy koncový. Vše můžeme vidět na obrázku 35. Rozložení stavů bylo poupraveno, aby se z tohoto grafu daly lépe vyčíst informace. Byli přemístěny jednotlivé stavy a nastavená možnost zobrazení všech hran.

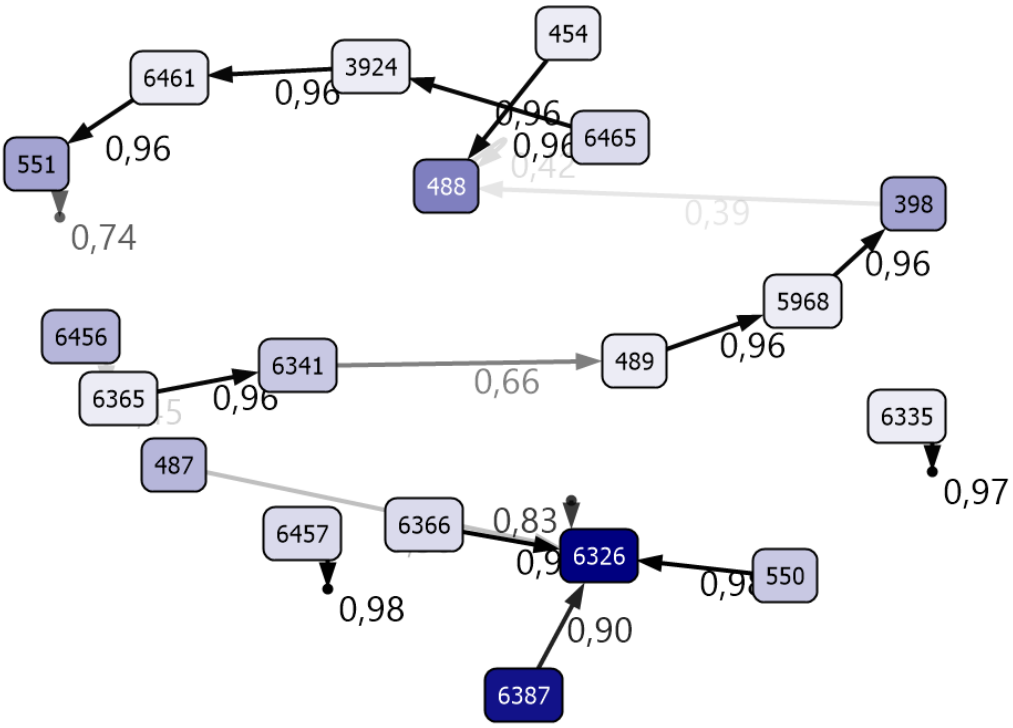
Když si ale zobrazíme *Cluster 2*, tak zjistíme, že s 83% pravděpodobností bude stav 6326 počátečním. Dalším důležitým zjištěním je, že pokud se budeme nacházet ve stavu 551, existuje 74% pravděpodobnost, že v tomto stavu zůstaneme, je tedy koncový. Žádné jiné vazby mezi stavy 6326 a 550 zde nejsou. Vše můžeme vidět na obrázku 36. Rozložení stavů je původní, bylo pouze nastaveno zobrazení dalších důležitějších hran v grafu o dva stupně oproti základnímu zobrazení.



Obrázek 34: Přehled vytvořených shluků a zobrazená část obsahu Cluster 1



Obrázek 35: Zobrazení pravděpodobnosti přechodů v Cluster 1



Obrázek 36: Zobrazené pravděpodobnosti přechodů v Cluster 2

7. Závěr

Hlavním cílem této diplomové práce bylo seznámení s problematikou pro dolování dat pomocí Microsoft SQL Server 2012 Standart Edition. Postupně bylo definováno co je to dolování dat a z jakých procesů se skládá, k čemu jej lze či nelze použít. Pro základní přehled bylo uvedeno, jaký algoritmus je vhodné použít pro získávání informací z určitých dat.

Tato práce byla zaměřená na vybrané algoritmy, které obsahuje Microsoft SQL Server 2012 Service Analysis. Byl získán základní přehled, jak algoritmy pracují a na jaká data je lze použít. U každého algoritmu jsme se také zaměřili na parametry, které obsahují, abychom je dokázali používat a věděli, k čemu slouží.

Od surových dat jsme se dostali až vytváření modelů pro dolování dat. Jako první bylo seznámení s daty a jejich zkoumání. Vstupní data byla v datových souborech, kdy musel být proveden import těchto dat do databáze. K tomu posloužila naprogramovaná aplikace, pomocí které byl proveden import dat do databáze.

Byli jsme seznámeni s prostředím Microsoft Visual Studio 2012 s nainstalovaným doplňkem Microsoft SQL Server Data Tools – Business Inteligence pro Visual Studio 2012. Díky tomu jsme pak mohli založit projekt pro dolování dat, ve kterém se nacházejí námi nakonfigurované modely. Byl vysvětlen celý postup nastavování modelů, co kde můžeme provádět a nastavovat, včetně základních nastavení projektu.

Významnou kapitolou byla analýza vytvořených modelů pro dolování dat. Ve které jsme se postupně věnovali prohlížení sestavených modelů, kdy pro každý použitý algoritmus jsme si kompletně nastudovali, jaké možnosti se nám nabízejí a především jaké informace se zde skrývají. Velice důležité byly zejména podkapitoly věnované vytváření nových odhadů, analýze úspěšnosti odhadů vytvořených modelů a v neposlední řadě analýze modelu pro sekvenční data. Naučili jsme se jak provádět nové odhady. Byla provedena analýza úspěšnosti odhadů vytvořených modelů, u kterých jsme si demonstrovali, jak je důležité věnovat se parametrům, které nalezneme v jednotlivých algoritmech. Pomocí úpravy některých parametrů se nám zpřesnily výsledky odhadů. Nakonec byla provedena analýza sekvenčního modelu, ve kterém jsme analyzovali obsah jednotlivých klastrů a sekvencí které obsahují.

8. Literatura

[1] L. Lacko, Business Intelligence v SQL Serveru 2008, Brno: Computer Press, a. s., 2009.

[2] „Data Mining (SSAS),“ Microsoft, 2012. [Online]. Available: http://technet.microsoft.com/en-us/library/bb510516.aspx. [Přístup získán 2 Leden 2014].

[3] J. Hynek a K. Ježek, „Automatická klasifikace dokumentů do tříd za použití metody Itemsets,“ [Online]. Available: http://textmining.zcu.cz/publications/Itemsets%20Datakon%202001.pdf. [Přístup získán 21 Duben 2014].

[4] P. Hájek, „Aplikace umělé inteligence,“ [Online]. Available: https://fim.uhk.cz/inkov/doc/SM_Hajek_10_06_2011.pdf. [Přístup získán 2 leden 2014].

[5] J. Kelbel a D. Šilhán, „Shluková analáza,“ [Online]. Available: http://www.fd.cvut.cz/personal/nagyivan/Projekty/Classification/ShlukovaAnalyza.pdf. [Přístup získán 20 Leden 2014].

[6] O. Štěpánková, „Dobývání a vizualizace znalostí,“ 5 Prosinec 2013. [Online]. Available: https://cw.felk.cvut.cz/wiki/_media/courses/a6m33dvz/09-shlukovani.pdf?cache=nocache. [Přístup získán 20 Leden 2014].

[7] P. Šenovský, Modelování rozhodovacích procesů, Ostrava: VŠB - Technická univerzita Ostrava, FBI, 2009.

[8] M. Litschmannová, Vybrané kapitoly z pravděpodobnosti, Ostrava: VŠB - Technická univerzita Ostrava, FEI, 2011.

[9] I. Vondrák, Umělá inteligence a neuronové sítě, Ostrava: VŠB - Technická univerzita Ostrava, 2001.

[10] J. Mohylová a V. Krajča, Zpracování biologických signálů, Ostrava: Editační středisko VŠB - Technická univerzita Ostrava, 2006.

[11] J. Demel, „Operační výzkum,“ [Online]. Available: http://kix.fsv.cvut.cz/~demel/ped/ov/ov.pdf. [Přístup získán 25 Duben 2014].

[12] A. Abraham a C. Grosan, „Evolving Intrusion Detection Systems,“ 2006. [Online]. Available: http://isda03.softcomputing.net/ids-chapter.pdf. [Přístup získán 25 3 2014].

[13] „.NET Framework Regular Expressions,“ Microsoft, 2014. [Online]. Available: http://msdn.microsoft.com/cs-cz/library/hs600312%28v=vs.110%29.aspx. [Přístup získán 25 Duben 2014].

[14] P. Berka, Dobývání znalostí z databází, Praha: Academia, 2003.

A. Testovací sestava

Tato diplomová práce byla vytvořena pro ilustraci na následující hardwarové sestavě, která je uvedena v tabulce 19. Samotná databáze byla umístěna ve virtuálním počítači, konfigurace je uvedena v tabulce 20.
Ve virtuálním počítači byl nainstalován následující software:

- Microsoft Visual Studio 2012 (Version 11.0.61030.00 Update 4)
- Microsoft SQL Server Data Tools – Business Inteligence pro Visual Studio 2012
- Microsoft Visual Studio 2013 (Version 12.0.21005.1 REL)
- Microsoft SQL Server 2012 Standart Edition – přehled nainstalovaných verzí se nachází na obrázku 37, výpis zobrazen pomocí Microsoft SQL Server Managament Studio.

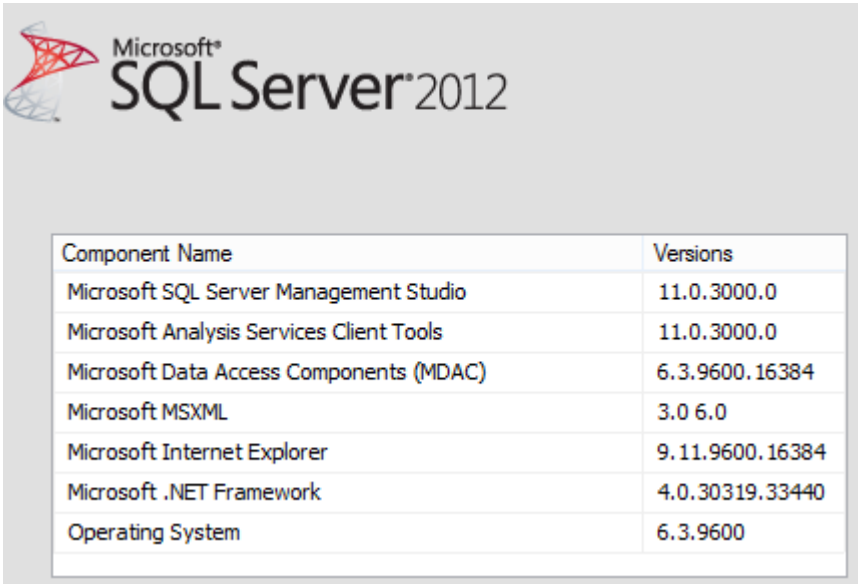
Operační systém fyzického počítače byl uložen na pevném disku RE4. Diskový prostor pro virtuální počítač byl uložen na disku FAEX a to z důvodu aby se tyto dva spuštěné systémy nějak neomezovaly.

Tabulka 19: Testovací sestava fyzického počítače

Procesor	Intel Core 2 Quad 9550 2,83 GHz
Základní deska	Gigabyte GA-EP45-UD3
Operační paměť	4x 2GB A-Data DDR2 1066MHz
Grafická karta	Gigabyte AMD Radeon HD6950 1GB (GV-R695OC-1GD)
Pevný disk	1TB Western Digital RE4 Raid (WD1003FBYX) 1TB Western Digital FAEX (WD1002FAEX) 1,5TB Western Digital GreenLine (WD15EADS)
Zdroj	Seasonic S12II-620, 620W
Operační systém	Windows 8.1 x64 Enterprise
Virtualizační software	VMware Workstation 10

Tabulka 20: Testovací sestava virtuálního počítače

Procesor	4 jádra
Operační pamět	4GB
Pevný disk	60GB – Operační systém a veškerý software 60GB – Pouze tabulky databáze
Operační systém	Windows 8.1 x64 Enterprise



Component Name	Versions
Microsoft SQL Server Management Studio	11.0.3000.0
Microsoft Analysis Services Client Tools	11.0.3000.0
Microsoft Data Access Components (MDAC)	6.3.9600.16384
Microsoft MSXML	3.0 6.0
Microsoft Internet Explorer	9.11.9600.16384
Microsoft .NET Framework	4.0.30319.33440
Operating System	6.3.9600

Obrázek 37: Microsoft SQL Server Management Studio, přehled nainstalovaných verzí

B. DVD-ROM

K této diplomové práci je přiloženo DVD medium, na kterém se nacházejí následující adresáře:

- **BI_DataMining –** vytvořený projekt v Microsoft Visual Studio 2012, obsahuje modely pro dolování dat.
- **Diplomová práce –** diplomová práce ve formátu PDF/A
- **SQL Skript** – obsahuje vytvořený SQL skript pro analýzu úspěšnosti odhadů modelů.
- **Testovaci data –** obsahuje testovací data, která byla použita v této diplomové práci.
- **WFA_DB_DataImport –** vytvořený projekt v Microsoft Visual Studio 2013 (zpětně kompatibilní s Microsoft Visual Studio 2012), obsahuje naprogramovanou aplikaci pro import testovacích dat do databáze.
- **WFA_AnalyzaSekvence –** vytvořený projekt v Microsoft Visual Studio 2013 (zpětně kompatibilní s Microsoft Visual Studio 2012), obsahuje naprogramovanou aplikaci pro analýzu sekvence.